COMMODORE & AMIGA nr 10



NR INDEKSU 355216
ISSN 0867-8022
Cena 10 000 zł
październik 1992 r.

MAGAZYN UŻYTKOWNIKÓW KOMPUTERÓW «COMMODORE»





KOMPUTERY ŁĄCZCIE SIĘ

TRAGEDIA GRACZA

TEXTOMAT PLUS

RAPORT: EDYTOR PL

ZOSTAŃ WŁAMYWACZEM

WYNIKI ANKIETY

Przywykliśmy już do obecności komputera w wielu miejscach, choć jeszcze nie tak dawno temu urządzenia te wywoływały sensację. Dziś klawiatury jakby spowszedniały — bez względu na miejsce, gdzie są zainstalowane.
Jak dotychczas, jako społeczeństwo, nie mamy w zasadzie powodów, aby uskarżać się na komputery. Prędzej czy później jednak maszyny te zaczną wchodzić do ogrodów zastrzeżonych obecnie wyłącznie dla człowieka. Przykład: roboty fabrykach samochodów. Komputer sterujący robotem odpowiedzialnym za zgrzewanie karoserii nie chodzi na zwolnienia, nie ma humorów, nie strajkuje, nie prosi o podwyżki, nie odnoszą się do niego żadne limity Kodeksu Pracy. Maszyna taka pracuje 24 godziny na dobę, nie jeździ na urlopy, nie ma przerw śniadaniowych, nie potrzebuje funduszu socjalnego — jednym słowem dla producenta jest to na pewno kosztowna ale pewna gratka obiecujaca duże zyski.
Z założenia komputer w biurze czy firmie powinien przyczyniać się do wzrostu dochodów i zautomatyzowania najbardziej uciążliwych i męczących prac. W Polsce przebiegało to nieco inaczej niż w innych państwach — u nas zaczęło się w zasadzie od zastępowania komputerami maszyn do pisania. Choć już wtedy było wiele programów, które mogłyby przyczynić się do szybkiego rozwoju i analizy strategicznych posunięć wielu firm, to jednak ówczesna kadra kierownicza wielokrotnie wolała wykazać się bilansem sporządzonym co prawda na komputerze lecz skontrolowanym liczydłem. W jednym ze znanych mi zakładów pracy służba finansowa odmówiła przyjęcia od pracownika bilansu sporządzonego za pomocą programu LOTUS 1-2-3 — trzeba było wrócić do starej Optimy z długim wałkiem.
Dziś na szczęście mamy ten okres za sobą; powoli dochodzimy do kolejnego etapu w którym szefowie i przełożeni będą zmuszeni do redukcji personelu, którego miejsca przejmie znacznie bardziej efektywny w działaniu komputer. Również i w tym wypadku są dwie strony medalu: z jednej strony wzrasta efektywność firmy i dyscyplina pracy (ludzie zagrożeni zaczynają pracę szanować), z drugiej zaś, w wielu wypadkach, komputer pozbawia całe rodziny podstaw do życia.
Patrząc na to z pesymistycznego punktu widzenia widzę wszak pewną nadzieję: należy jak najszybciej z komputerem się zapoznać i oswoić. Jeśli bliska wizja komputerów zastępujących urzędników niebawem zacznie się sprawdzać, to na pewno będą potrzebni ludzie do ich programowania, konserwacji i obsługi.

KLAUDIUSZ DYBOWSKI

M·E·N·U

AMIGA

C-64

ORAZ

magazyn użytkowników komputerów «COMMODORE»



Redaktor naczelny: KLAUDIUSZ DYBOWSKI
Sekretarz redakcji: CHRISTIAN GRZENKOWICZ
Opracowanie graficzne: JOLANTA PRZEŹDZIECKA
Zdjęcia: JERZY STOKOWSKI

Stali współpracownicy: ANDRZEJ BOBEK (szef działu Amigi), RAFAŁ BORZYŃSKI, PIOTR CERKIEWNIK, BARTŁOMIEJ DRAMCZYK, JERZY DUDEK, MARIUSZ FERDYN, PAWEŁ GALAS, BARTŁOMIEJ KACHNIARZ, WOJCIECH KAZIMIERCZAK, ROBERT KULIŚ, PIOTR LISZEWSKI, RAFAŁ PIASEK, OLAF PRZYBYSZEWSKI, BARTOSZ SMAGA

Redakcja: ul. Wasilkowskiego 7, 02-776 WARSZAWA tel./BBS: 643-1840
Kontakt z Czytelnikami: pon-pt w godzinach 10.00-17.00
Wydawca: Spółdzielnia „Bajtek" ul. Wspólna 61 00-687 Warszawa tel./fax: 21-12-05
Skład i druk: Przedsiębiorstwo Wydawniczo-Poligraficzne „GRYF", Sp. Akc. Ciechanów
Korekta: KRYSTYNA WYDURSKA, MARIA GOŹDZIEWSKA
Nr zlecenia: 52302
Nakład: 72 tys. egzemplarzy



OTWÓRZ OKIENKO...

Graficzne środowisko Amigi daje nam do ręki niezwykłe narzędzie, jakim są okienka. Każdy program może dysponować własnym oknem, w którym umieszcza wyniki swego działania i nie ingeruje dzięki temu w obrazy tworzone przez inne programy. Jak okna wyglądają i jakie mają właściwości, można przekonać się oglądając choćby Workbench.

Innym narzędziem są ekrany, które mogą mieć różne rozdzielczości, palety i liczbę kolorów. Ekrany możemy chować jeden pod drugi, czy też „zsuwać" je w dół.

Zacznijmy od ekranów. Aby nasz program miał własny ekran, musi go sobie otworzyć. Służy do tego funkcja OpenScreen() znajdująca się w bibliotece Intuition. Daną wejściową dla niej jest wskaźnik do struktury NewScreen, a wynikiem wskaźnik do struktury Screen otwartego ekranu. W skład struktury NewScreen wchodzą pola:

LeftEdge, TopEdge — współrzędne lewego górnego rogu ekranu;
Width, Height — szerokość i wysokość ekranu;
Depth — liczba planów bitowych, a co za tym idzie liczba kolorów jest równa 2^Depth;
DetailPen — numer koloru dla tekstu w listwie tytułowej ekranu;
BlockPen — numer koloru listwy tytułowej;
ViewModes — tryb wyświetlania np.:
- *HIRES* — wysoka rozdzielczość pozioma, 640 punktów w linii;
- *INTERLACE* — podwójna rozdzielczość pionowa, 512 linii;
- *SPRITES* — używane duszki;
- *TYPE* — typ ekranu:
- *CUSTOMSCREEN* — nasz własny ekran, a nie systemowy;
- *CUSTOMBITMAP* — własna mapa bitowa;
- *SCREENBEHIND* — ekran jest otwierany pod wszystkimi otwartymi ekranami;

Font — wskaźnik do struktury TextAttr opisującej czcionkę używaną we wszystkich napisach; jeśli Font=NULL wtedy czcionka jest standardowa;
DefaultTitle — wskaźnik do łańcucha tekstu zakończonego zerem, zawierającego tytuł ekranu;
Gadgets — wskaźnik do listy *gadgetów* (przełączników) lub NULL gdy ich nie ma;
CustomBitMap — wskaźnik do własnej mapy bitowej jeżeli było ustawione CUSTOMBITMAP.

Jeżeli w trakcie otwierania wystąpi błąd, to funkcja zwróci wartość NULL zamiast wskaźnika do struktury **Screen**. Wartości **ViewModes** i **Type** są sumami wybranych opcji. Każdy otwarty ekran przed zakończeniem programu powinien zostać zamknięty, co robimy za pomocą procedury **CloseScreen()**, dla której daną jest wskaźnik do struktury **Screen** zamykanego ekranu.

Gdy już mamy własny ekran (albo chcemy wykorzystać do tego ekran Workbencha), możemy otworzyć nasze okno. Można tego dokonać za pomocą funkcji **OpenWindow()** ze strukturą **NewWindow** jako daną. W skład struktury **NewWindow** wchodzą: **LeftEdge, TopEdge, Width, Height, DetailPen, BlockPen** — o znaczeniu takim jak dla ekranu; **IDCMPFlags** — znaczniki określające jakie zdarzenia będą przekazywane z okna do programu poprzez port komunikatów; **Flags** — znaczniki opisujące cechy okna; **FirstGadget** — wskaźnik do pierwszego z listy *gadgetów* dołączonych do okna; **CheckMark** — wskaźnik do struktury Image opisującej symbol graficzny, służący do zaznaczenia ustawionej opcji w menu (jeśli równy NULL to przyjmowany jest standardowy; **Title** — wskaźnik do teksu z tytułem okna; **Screen** — wskaźnik do struktury **Screen** ekranu na którym otwieramy okno, gdy jest ono typu *CUSTOMSCREEN*; **BitMap** — wskaźnik do mapy bitowej, jeżeli stosujemy własną; **MinWidth,MinHeight** — minimalna szerokość i wysokość okna; **MaxWidth,MaxHeight** — maksymalna szerokość i wysokość; **Type** — typ ekranu na którym otwieramy okno:

WBENCHSCREEN — systemowy ekran Workbencha;
CUSTOMSCREEN — własny ekran otwarty wcześniej.

Zmienna **Flags** zawiera znaczniki:

WINDOWDRAG — pasiasta listwa tytułowa do przesuwania okna;
WINDOWDEPTH — przełączniki do zmiany kolejności zasłaniania okien;
WINDOWCLOSE — wyłącznik okna;
WINDOWSIZING — element do zmiany rozmiarów okna;
SIZEBRIGHT — rezerwacja miejsca przy pewnym marginesie, np. dla suwaków;
SIZEBBOTOM — jak wyżej lecz dla dolnego marginesu;
GIMMEZEROZERO — przesunięcie początku układu współrzędnych z lewego górnego rogu okna, do punktu poniżej listwy tytułowej i na prawo od lewej ramki;
BACKDROP — okno będzie tłem dla innych okien, których nie będzie można pod nie schować;
BORDERLESS — okno bez ramek;
ACTIVATE — okno otwierane jako aktywne;
REPORTMOUSE — umożliwia odczyt współrzędnych myszy w stosunku do lewego górnego rogu okna;
RMBTRAP — umożliwia przechwycenie przyciśnięcia prawego przycisku myszy;
SUPER_BITMAP — stosujemy własną mapę bitową

Znaczniki w zmiennej **IDCMPFlags** to np.:

CLOSEWINDOW — przesyłana będzie informacja o naciśnięciu wyłącznika okna;
MOUSEBUTTONS — przyciśnięcie bądź zwolnienie przycisków myszy (jeżeli we **Flags** nie jest ustawione RMBTRAP, to tylko lewego);
MOUSEMOWE — przesyłana będzie informacja o przesunięciach myszy;
ACTIVEWINDOW — informowanie o aktywacji okna;
INACTIVEWINDOW — informowanie o dezaktywacji okna. Innymi informacjami które mogą być przesyłane do programu są np.: naciśnięcie bądź zwolnienie umieszczonych przez program w oknie *gadgetów*, klawiszy na klawiaturze, zmiana wymiarów okna, wybranie opcji z menu itd...

Wynikiem otrzymywanym z funkcji **OpenWindow()** jest wskaźnik do struktury Window otwartego okna lub NULL, gdy wystąpił błąd. W przypadku okien również musimy pamiętać o ich zamknięciu przed zakończeniem programu. Okno zamykamy procedurą **CloseWindow()** ze wskaźnikiem do struktury Window zamykanego okna jako daną wejściową.

Gdy mamy już własne okno, możemy wreszcie w nim rysować czy pisać. Dokonujemy tego korzystając z funkcji biblioteki **Graphics**. Aby wiadomo było, w którym oknie należy wyświetlać np. grafikę, procedurom tym niezbędny jest wskaźnik do struktury RastPort naszego okna. Znajdujemy go w składowej struktury Window pod nazwą RPort.

Na początek kilka zupełnie podstawowych funkcji. Kursor graficzny przesuwamy przez Move(*RastPort,x,y), linię od bieżącej pozycji kursora do zadanej rysujemy procedurą Draw(*RastPort,x,y). Tekst umieszczamy funkcją Text (*RastPort,Tekst,Długość), przy czym lewy dolny róg pola pierwszego znaku umieszczany jest w bieżącej pozycji kursora graficznego. Pojedynczy punkt stawiamy za pomocą funkcji WritePixel(*RastPort,x,y), która zwraca wartość 0 gdy nie było błędu lub — 1 gdy punkt jest poza oknem. Kolor, którym rysuje-



my ustawiamy poprzez SetAPen(*RastPort,nr_koloru), tło ma kolor nr 0. Możemy również narysować wypełniony prostokąt procedurą RectFill(* RastPort,xmin,ymin,xmax,ymax), dla koloru nr 0 można ją wykorzystać do kasowania zawartości obszaru lub nawet całego okna.

Procedur graficznych, obsługujących różne przełączniki, manipulujących oknami i ekranami jest kilkadziesiąt. Przedstawiłem tutaj tylko wierzchołek góry lodowej, ale można już na nim stanąć. Na koniec program, który pokazuje jak to wykorzystać w praktyce.

MARCIN JAZGAR

```
/* Screen & Window Demo */

#include <exec/types.h>            /*dołączenie potrzebnych plików*/
#include <intuition/intuition.h>
#include <intuition/intuitionbase.h>
#include <functions.h>
#include <graphics/gfxbase.h>

struct IntuitionBase *IntuitionBase; /*wskaźnik do biblioteki intuition*/
struct GfxBase *GfxBase;           /*wskaźnik do biblioteki graphics*/
struct Screen *Ekran;              /*wskaźnik do struktury Screen*/
struct Window *Okno;               /*wskaźnik do struktury Window*/
struct RastPort *RP;               /*wskaźnik do struktury RastPort*/
struct IntuiMessage *Msg;          /*wskaźnik do struktury komunikatu*/

struct NewScreen NowyEkran =       /*struktura NewScreen*/
{
  0,0,                     /*wsp.lewego górnego rogu*/
  320,256,                 /*szerokość, wysokość*/
  3,                       /*ilość planów - 8 kolorów*/
  4,3,                     /*DetailPen,BlockPen*/
  0,                       /*standardowy tryb wyświetlania 320*256*/
  CUSTOMSCREEN,            /*typ ekranu*/
  NULL,                    /*standardowa czcionka*/
  "Ekran - Demo",          /*tytuł ekranu*/
  NULL,                    /*brak własnych przełączników*/
  NULL                     /*brak własnej mapy bitowej*/
};

struct NewWindow NoweOkno =        /*struktura NewWindow*/
{
  50,50,                   /*wsp.lewego górnego rogu*/
  200,150,                 /*szerokość, wysokość*/
  6,2,                     /*DetailPen,BlockPen*/
CLOSEWINDOW,               /*przekazywana będzie informacja*/
                           /*o użyciu wyłącznika okna*/
WINDOWDRAG | WINDOWCLOSE | WINDOWSIZING | GIMMEZEROZERO | ACTIVATE,
                           /*opis zmiennej Flags w tekście*/
NULL,                      /*brak własnych przełączników*/
NULL,                      /*standardowy znacznik menu*/
"Okno !",                  /*tytuł*/
NULL,                      /*wskaźnik do strukt. Screen*/
                           /*będzie wstawiony po otwarciu ekranu*/
NULL,                      /*brak własnej mapy bitowej*/
50,50,                     /*min. szerokość i wysokość*/
300,200,                   /*max. szerokość i wysokość*/
CUSTOMSCREEN               /*okno na własnym ekranie*/
};

main()                 /*procedura główna*/
{                          /*otwarcie potrzebnych bibliotek*/
GfxBase = (struct GfxBase *)OpenLibrary("graphics.library",0);
if (GfxBase==NULL) Koniec();
IntuitionBase = (struct IntuitionBase *)OpenLibrary("intuition.library",0);
if (IntuitionBase==NULL) Koniec();
Ekran = OpenScreen(&NowyEkran);    /*otwarcie ekranu*/
if (Ekran==NULL) Koniec();
NoweOkno.Screen = Ekran;   /*wskaźnik do ekranu wstawiany do NewWindow*/
Okno = OpenWindow(&NoweOkno);      /*otwarcie okna*/
if (Okno==NULL) Koniec();
RP=Okno->RPort;            /*wskaźnik do RastPort'u okna*/
Demo();
for (;;)                   /*oczekiwanie na naciśnięcie wyłącznika*/
 {                         /*okna*/
  Msg = WaitPort(Okno->UserPort);
  Msg = GetMsg(Okno->UserPort);
  if (Msg->Class == CLOSEWINDOW) Koniec();
  ReplyMsg(Msg);
  }
}

Koniec()              /*zamknięcie wszystkiego co było otwarte*/
{                     /*i wyjście z programu*/
if (Okno) CloseWindow(Okno);
if (Ekran) CloseScreen(Ekran);
if (GfxBase) CloseLibrary(GfxBase);
if (IntuitionBase) CloseLibrary(IntuitionBase);
exit(0);
}

Demo()
{
int i,k;

for (i=0;i<=70;i+=10)
 {
  SetAPen(RP,i/10+1);
  RectFill(RP,20+i,10+i/2,167-i,130-i/2);
 }
for (i=20;i<167;i+=21)
 for (k=1;k<=7;k++)
  {

   SetAPen(RP,k);
   Move(RP,i+3*(k-1),10);
   Draw(RP,187-(i+3*(k-1)),130);
  }
SetAPen(RP,1);
Move(RP,58,74);
Text(RP,"Graf-Demo",9);
}

/*  Kompilator   Aztec C
  kompilacja     cc -s +l nazwa.c
  konsolidacja   ln nazwa.o -lc32 */
```



KABLOLOGIA cz. II

Podłączenie monitora kolorowego ATARI ST do Amigi

Często bywa tak, że peryferiów od starego komputera nie uda się sprzedać i stoją one bezużytecznie w charakterze mebli. Zastanawiacie się, czy można je podłączyć do nowego nabytku. Tu przyjdzie Wam z pomocą seria artykułów z cyklu „Kablologia". W dzisiejszym odcinku schemat podłączenia monitora kolorowego od ATARI ST do Amigi. W tym miejscu należy się wyjaśnienie: nie jest możliwe podłączenie w równie prosty sposób monitora monochromatycznego ATARI (np. SM 124 czy SM 125). Powodem tego są różne częstotliwości synchronizacji pionowej i poziomej.

Schemat kabelka przedstawiony jest na rysunku. Widoczne dwa wtyki typu Cinch nie muszą być podłączone, jeżeli nie zamierzamy wykorzystać wewnętrznego głośnika monitora (tylko podłączamy Amigę do wzmacniacza). Jeśli mamy kłopoty ze zdobyciem gniazda monitora ATARI, najprościej jest odlutować wtyczkę monitora i podłączyć przewody wprost do wtyczki Amigi (23-stykowy żeński Canon). Przed odlutowaniem należy zapisać kolory przewodów i numery styków, do których były przyłączone, gdyż później możemy mieć problemy z identyfikacją!

JERZY DUDEK

KOMPUTERY ŁĄCZCIE SIĘ! cz. II

Połączenie Amigi z IBM PC

W tym artykule kontynujemy tematykę łączenia komputerów ze sobą. Poprzednio omówiliśmy połączenie dwóch Amig ze sobą. Teraz pora na połączenie Amigi z IBM PC. Ponieważ niektóre PC-ety mają gniazdo RS-232 25-stykowe, to do podłączenia Amigi konieczny jest przewód opisany w poprzednim artykule. Jednak część IBM-ów i wszystkie laptopy mają łącze 9-stykowe. Wszystkie teoretyczne uwagi opisane poprzednio dotyczą również tego kabla, dlatego nie będziemy powtarzać zamieszczonych tam informacji.

Do wykonania kabelka konieczne są:

— wtyczka typu Canon 9-stykowa, żeńska

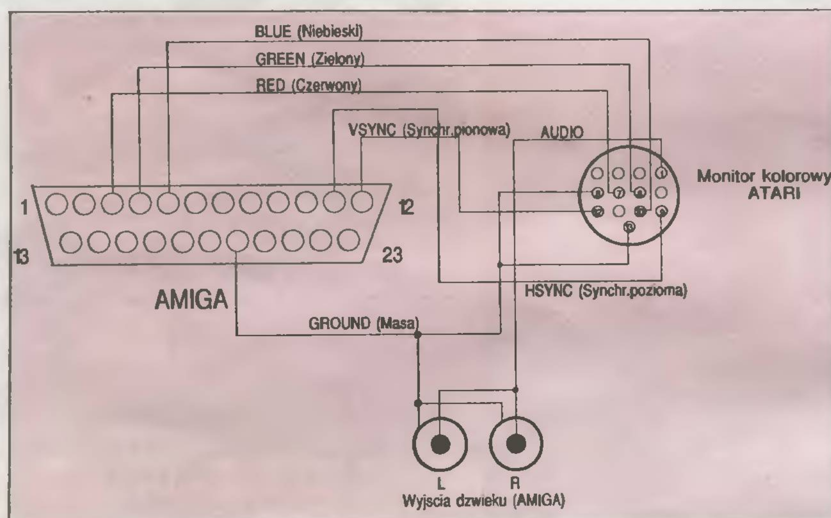
— wtyczka typu Canon 25-stykowa, żeńska

— przewód siedmiożyłowy o dowolnej długości, może być nieekranowany.

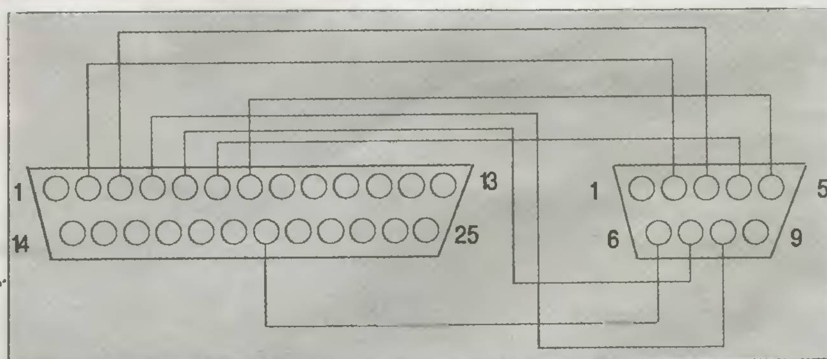
Schemat połączeń przedstawiono na rysunku. Po wykonaniu przewodu i sprawdzeniu poprawności połączeń możemy wykonać próbną transmisję. Opis wykonywania transmisji na IBM PC za pośrednictwem RS-232 jest opisany w „Bajtku" nr 6/92.

Generalnie należy pamiętać, że parametry transmisji ustawione na obu komputerach MUSZĄ być identyczne, aby uniknąć przekłamań. Na Amidze z poziomu DOS-u należy posłużyć się urządzeniem ser:.

JERZY DUDEK



Rys. 1. Połączenie monitora kolorowego ATARI do Amigi



Rys. 2 Schemat kabla do połączenia Amigi z IBM PC poprzez łącze szeregowe RS-232

ASEMBLER 68000
(CZĘŚĆ III)

Ostatnio dowiedzieliśmy się, w jaki sposób można otworzyć okienko na ekranie i wydrukować na nim swój własny tekst. Dzisiaj nauczymy się, jak odczytywać znaki z klawiatury i umieszczać je na oknie.

Odczyt klawiatury jest bardzo prosty. Musimy jednak wcześniej zapoznać się z procedurą „READ" (offset –$2a), umieszczoną w bibliotece „dos.library".

Przed wywołaniem procedury musimy w rejestrach D1, D2 i D3 umieścić odpowiednie wartości:

w **D1** wartość, którą zwrócił nam system po otwarciu okna za pomocą procedury „OPEN" (tzw. Conhandle),
w **D2** adres, pod jakim będą umieszczane kody ASCII kolejno wprowadzanych znaków z klawiatury,
w **D3** liczba znaków, które mają zostać odczytane z klawiatury.

Poniżej znajduje się krótki podprogram, który odczytuje i „zapamiętuje" wciskane klawisze. Należy pamiętać o wcześniejszym otworzeniu okienka (za pomocą procedury „OPEN") i umieszczeniu wartości z rejestru D0 (conhandle) pod etykietą „OknoHD". Przed wywołaniem podprogramu w rejestrze D3 powinna znaleźć się liczba określająca, ile znaków ma być odczytanych znaków.

```
Read=–$02a
Odczyt Klawiatury:
        move.l #TuZnaki,d2    ; początek obszaru pamięci pod
                              ; dane wprowadzane z klawiatury
        move.l DosBase,a6     ; adres bazowy biblioteki „DOS"
        move.l OknoHD,d1      ; conhandle w rejestrze D1
        jsr    Read(a6)       ; wywołanie procedury „READ"
        rts
TuZnaki:
        BLK.B 80,0            ; zarezerwowanie 80 bajtów pamięci
                              ; dla danych z klawiatury
```

Odczyt klawiatury będzie trwał do momentu, aż zostanie wciśnięty klawisz ‹RETURN›. Jeżeli liczba wprowadzonych przez nas znaków będzie większa od wartości umieszczonej w rejestrze D3, „zapamiętane" zostaną tylko pierwsze wprowadzone znaki. Kody ASCII wprowadzonych znaków będą zapisywane od adresu określonego etykietą „TuZnaki".

Bardzo często zależy nam, aby program czekał pewien określony odcinek czasu na podanie przez nas jakiegoś parametru. Możemy użyć procedury o nazwie „WAITFORCHAR" (offset –$cc) umieszczonej w bibliotece „dos.library". Po jej wywołaniu w rejestrze D0 system zwróci nam wartość określającą, czy został wciśnięty jakikolwiek klawisz. Wciśnięcie klawisza spowoduje pojawienie się w rejestrze D0 wartości różnej od zera. Jeżeli jednak znajdziemy tam 0, to jest to informacja od systemu, że w zadanym przez nas czasie żaden klawisz nie został wciśnięty.

Wywołanie procedury wymaga od nas podania dwóch wartości:

w **D1** „conhandle" okna, z którego ma zostać dokonany odczyt klawiatury,
w **D2** czas (w mikrosekundach) oczekiwania na wciśnięcie klawisza.

Poniżej podaję krótki podprogram, który czeka jedną sekundę na wciśnięcie klawisza:

```
WaitForChar=–$0cc
Czekamy:
    move.l ConHD,d1
    move.l #1000000,d2
    move.l DosBase,a6
    jsr    WaitForChar(a6)
    tst.l  d0
    rts
```

Na końcu podprogramu umieszczona jest instrukcja TST, która pozwala nam na użycie w programie głównym instrukcji BEQ lub BNE. Testuje ona zawartość rejestru D0 i w zależności od jego zawartości ustawia odpowiednie kody warunkowe. Pozwoli nam to na sprawdzenie, czy klawisz został wciśnięty, np.:

```
bsr      Czekamy              ; czekamy 1 sekundę na
                              ; wciśnięcie klawisza
beq.w    NieWciśnięto         ; jeżeli w D0 znaleziono 0
                              ; to ma wykonać skok
. . .                         ; dalsza część programu
. . .
NieWciśnięto:                 ; program, który zadziała jeżeli nie
. . .                         ; nie wciśnięto żadnego klawisza
rts                           ; wyjście
```

Teraz, wyobraźmy sobie, że chcemy porównać wpisane z klawiatury słowo ze słowem, które znajduje się w pamięci. Może to być pomocne np. jeżeli chcemy, aby nasz program uruchamiał się na podane hasło. Poniżej znajduje się program, który porównuje dwa słowa znajdujące się w pamięci. Nasze hasło będzie umieszczone od adresu określonego etykietą „Hasło". Wprowadzony z klawiatury tekst będzie rozpoczynał się od adresu określonego etykietą „TuZnaki". Podane przez nas słowo, jak i hasło musi zawsze kończyć się zerem.

```
PORÓWNAĆ:
    lea   HASŁO(pc),a0        ; początkowy adres hasła
    lea   TuZnaki(pc),a1      ; początkowy adres słowa

PORÓWNAĆpętla1:
    move.b (a0)+,d0           ; litera hasła do DO
    move.b (a1)+,d1           ; litera słowa do D1
    cmp.b d0,d1               ; porównanie litery hasła z
                              ; literą słowa
    bne.w RÓŻNESŁOWA          ; wykona skok do etykiety
```

```
                          ; „RÓŻNESŁOWA" jeżeli słowa nie
                          ; będą takie same
   tst.b d0               ; sprawdzenie czy to nie jest
   bne.w PORÓWNANIEpętla1 ; koniec hasła, jeśli nie
                          ; to porównujemy następne litery
   ; jeżeli zostało podane prawidłowe hasło, to można
   ; przejść do wykonania dalszej części programu
   ...                    ; dalsza część programu
RÓŻNESŁOWA:               ; podane słowo nie jest takie
                          ; samo jak nasze hasło
   rts                    ; opuszczamy nasz program
HASŁO: dc.b 'C&A',0 ; to jest nasze hasło
   even                   ; instrukcja asemblera
                          ; ustawiająca parzysty adres
```

W programie pojawia się instrukcja CMP, która porównuje dwie wartości znajdujące się w rejestrze D0 i D1. Porównanie to polega na odjęciu od siebie zawartości obu rejestrów. Operacja ma rozmiar jednego bajtu (CMP.B) i oznacza to, że bajt z rejestru D0 zostanie odjęty od bajtu rejestru D1 i na podstawie tak uzyskanego wyniku zostaną ustawione kody warunkowe.

Za miesiąc poznamy w jaki sposób można odczytywać i zapisywać pliki na dyskietce. Wcześniej jednak musimy poznać trzy procedury systemu operacyjnego służące do rezerwowania pamięci. Po co to nam będzie potrzebne? Chcąc odczytać dane z dyskietki musimy podać systemowi adres, pod jakim dane te mają być umieszczone. Bardzo często trudno jest odnaleźć miejsce w pamięci, z którego nie korzystałby jakiś program, a procedury te pomogą nam takie miejsce odnaleźć.

Procedury, które zajmują się rezerwowaniem pamięci, znajdują się w bibliotece „exec.library" i są to: „AllocMem" (offset -$c6) i „AllocAbs" (offset-$cc). Procedura „AllocMem" szuka w pamięci wolnego obszaru o zadanej przez nas długości i podaje nam jego adres początkowy. W rejestrze D0 musi znajdować się wartość określająca, jaką długość obszaru (w bajtach) chcemy zarezerwować, a w D1 rodzaj rezerwowanej pamięci. W rejestrze D1 możemy podać kilka różnych wartości:

$00002 — obszar będzie leżał w dolnych 512kB, tzw.CHIP-RAM
$00004 — obszar będzie leżał w FAST-RAM,
$10002 — obszar w CHIP-RAM i zostanie wyzerowany,
$10004 — obszar w FAST-RAM i zostanie wyzerowany,
$20000 — otrzymujemy największy dostępny blok pamięci.

Po powrocie z procedury „AllocMem" w rejestrze D0 znajdziemy adres początku obszaru zarezerwowanej pamięci. Jeśli wartość w D0 będzie równa 0, to obszar o zadanej przez nas długości nie może zostać zarezerwowany.

```
ExecBase=4
AllocMem=-$c6
...
move.l #880,d0           ;ma zarezerwować 880 bajtów pamięci
move.l #$10002,d1        ;obszar pamięci ma być
                         ;wyczyszczony i umieszczony w
                         ;CHIP-RAM
move.l $4,a6             ;adres tablicy skoków Exec do A6
jsr    AllocMem(a6)      ;wywołanie podprogramu
move.l d0,AdresPamięci   ;zapamiętanie adresu
                         ;zarezerwowanej pamięci
beq    koniec            ;wyjście z programu jeśli nie został
                         ;zarezerwowany obszar pamięci

...                      ;dalsza część naszego programu
koniec:
rts
   AdresPamięci:
   blk.l 1               ;miejsce na zapisanie adresu
                         ;zarezerwowanego obszaru
```

Drugim sposobem na zarezerwowanie pamięci jest użycie procedury „AllocAbs". Działa on trochę inaczej niż „AllocMem", a mianowicie rezerwuje obszar pamięci o początku i długości przez nas podanej. Podajemy dwie wartości wejściowe: w rejestrze A1 adres początku rezerwowanej pamięci i w D0 jej długość. Tak samo jak poprzednio wartość zwrócona nam przez system w rejestrze D0 określa, czy pamięć została zarezerwowana. Tak wygląda przykładowy program:

```
ExecBase=4
AllocAbs=-$cc
   ...
   move.l #880,d0        ;wielkość rezerwowanej pamięci
   lea adrespamięci, a1  ;początek rezerwowanego obszaru
   move.l $4,a6
   jsr AllocAbs(a6)      ;uruchomienie podprogramu
   tst.l d0              ;sprawdzamy, czy pamięć została
                         ;zarezerwowana
   beq koniec            ;jeśli nie, to opuszczamy nasz program
   ...                   ;jeśli tak, to dalsza część programu
koniec:
   rts
adres pamięci:           ;początek rezerwowanej pamięci
```

Należy pamiętać, że jeśli powracamy z naszego programu do CLI lub Workbench'a, musimy zwolnić zarezerwowaną przez nas pamięć. Służy do tego procedura o nazwie „FreeMem" (offset −$d2).

Wartości wejściowe są takie same jak w przypadku „AllocAbs": w A1 podajemy adres początku zarezerwowanego obszaru i w D0 jego długość. Jeżeli spróbujemy zwolnić obszar pamięci, który wcześniej nie został zarezerwowany, możemy spowodować „zawieszenie się" systemu. Po wywołaniu „FreeMem" musimy sprawdzić zawartość rejestru D0, wartość, którą tam znajdziemy, poinformuje nas czy proces zwolnienia zarezerwowanego obszaru pamięci przebiegł pomyślnie (0 — obszar nie został zwolniony).

```
ExecBase=4
FreeMem=-$d2
   ...
   move.l $880, d0       ;wielkość zwalnianego obszaru
   lea adrespamięci, a1  ;adres początku obszaru
   move.l 4,a6
   jsr FreeMem(a6)       ;uruchomienie FreeMem
   tst.l d0              ;testujemy zawartość D0
   beq Koniec            ;jeśli=0 to zwolnienie
                         ;obszaru nie nastąpiło
   ...
```

Zarezerwowanie pamięci za pomocą procedury systemu operacyjnego gwarantuje nam, że w pracy w multitaskingu inny, działający równolegle program nie będzie korzystał z zarezerwowanej pamięci i tym samym nie będzie wprowadzał zakłóceń w pracy naszego programu.



BARTOSZ SMAGA

PAMIĘĆ CZY SKLEROZA?

TEST ROZSZERZENIA PAMIĘCI DO 2.5 MB

Podstawowym i najczęściej spotykanym dodatkiem do Amigi jest rozszerzenie pamięci. Zwykle jest to tzw. „małe" rozszerzenie (czyli 0.5 MB) wkładane pod spód komputera. Do gier jest ono w zupełności wystarczające, jednak do poważniejszej pracy absolutnie się nie nadaje. Wiele programów możemy tylko uruchomić (np. Scenery Animator II, czy Pixel 3D), a w wielu innych część opcji jest niedostępna lub mamy poważne ograniczenia w korzystaniu z możliwości przez nie oferowanych.

Tak więc jedyną alternatywą jest zakup rozszerzenia oferującego nam większą ilość pamięci. Musimy się tylko zdecydować, czy będzie to rozszerzenie instalowane w środku komputera (SLOW), czy montowane z boku (FAST). Najczęściej decydujemy się na to pierwsze rozwiązanie, szczególnie ze względów finansowych — jest ono tańsze.

Chcąc ułatwić Wam wybór, pragnę przedstawić test modułu rozszerzenia pamięci do 2.5 MB firmy Alfa Data o symbolu RA5-2MB (wyposażone w zegar). Otrzymaliśmy je do testowania dzięki uprzejmości firmy „PROABIT" (dawniej Adenek) będącej jego dystrybutorem.

MONTAŻ

Na pierwszy rzut oka rozszerzenie prezentuje się bardzo efektownie. Ładne, eleganckie opakowanie, wraz z informacją w czterech językach o jego zawartości, a także danymi technicznymi. Po jego otwarciu wyjmujemy rozszerzenie (zapakowane w antystatyczną torebkę), gwarancję, instrukcję obsługi a także dyskietkę i kolorową reklamówkę innych produktów firmy. Niestety, mimo zapewnień na opakowaniu, że zawiera ono instrukcję w czterech językach (niemiecki, angielski, francuski i włoski), jest ona wyłącznie po niemiecku, co dla osób nie znających tego języka może być pewnym problemem (instrukcja zawiera bardzo ważne informacje na temat montażu i ustawienia konfiguracji rozszerzenia). Z instrukcji można się dowiedzieć m.in., że moduł jest produkowany w dwóch wersjach: pierwszej, podstawowej umożliwiającej rozszerzenie wyłącznie pamięci SLOW (testowane) i drugiej, rozszerzającej jednocześnie pamięć CHIP i SLOW. Różnica polega jedynie na dodaniu specjalnej podstawki pod mikroprocesor.

Z torebki wyjmujemy płytkę rozszerzenia, podstawkę pod Gary'ego, kabelek łączący te płytki i drugi kabelek z przełącznikiem na końcu, pozwalający na wyłączenie rozszerzenia bez konieczności jego wyjmowania. Montaż wymaga otwarcia komputera (co wiąże się z utratą gwarancji!), a także odrobiny cierpliwości i delikatnych palców.



PRODUCENT: ALFA-DATA
DYSTRYBUTOR: PROABIT (dawniej Adenek), 05-500 Raszyn k/Warszawy, ul. Mickiewicza 14, tel. 560891.
CENA: ok. 1.800.000 zł (225 DM)

Po dokonaniu tej czynności należy ustawić konfigurację rozszerzenia (czy będzie pracować jako 0.5 MB, 1 MB, 1.5 MB czy 2.0 MB — wszystko jest wyjaśnione w instrukcji). Ustawiłem je jako pełne 2 MB.

PRACA

Po zmontowaniu komputera i jego włączeniu pełna ulga — działa! Na ekranie monitora pojawiła się dobrze znana, tzw. „żebrząca rączka". Na początek uruchomiłem program testujący znajdujący się na dołączonej dyskietce. Po chwili oczekiwania wyświetlił on informację, że wszystko jest w porządku — rozszerzenie zostało zainstalowane poprawnie.

Jak wiadomo, pamięć SLOW może mieć tylko 1792 KB, brakujące (do pełnych 2 MB) 256 KB znajdować się będzie od adresu $F40000 i zostaje tam zainstalowane przez programik EFRAM — Extra Fast RAM (również na dyskietce).

W czasie użytkowania rozszerzenia (2 tygodnie) starałem się przetestować je jak najdokładniej. Po tym okresie nasuwają mi się następujące wnioski:

- wszystkie gry i dema, które mam, pracują poprawnie i nie wykazują żadnych odstępstw do normy;
- niestety nie wszystkie programy użytkowe pracują poprawnie, niektóre mimo poprawnej pracy na innych, mają pewne problemy z tym rozszerzeniem. Występują błędy z allokacją pamięci. Zdarzają się przypadki, że długość wolnego, największego bloku do zaalokowania z 1.7 MB zmniejsza się po wgraniu danego programu (Directory Opus czy Lha) do np. 20 bajtów! (oczywiście w obszarze pamięci SLOW). Mimo podejrzeń o uszkodzenie układów scalonych, wszystkie programy testujące nie wykazują żadnych błędów;
- zła praca zegara — „uwielbia" się przestawiać, a czasami przyjmuje wartości nierozpoznawane przez system;
- wyłączenie rozszerzenia (specjalnym wyłącznikiem) jest poprawnie rozpoznawane przez system (po resecie), natomiast jego włączenie wymaga wyłączenia komputera na czas ok. 30 sekund (dopiero wtedy jest rozpoznawane); jest to jednak już, wina samej Amigi, a właściwie systemu operacyjnego
- próby zmiany konfiguracji rozszerzenia (np. żeby pracowało jako 1 MB) nie dają żadnego efektu. Pracuje tak, że mamy „wszystko albo nic".

PODSUMOWANIE

Moduł RA5-2MB jest interesującą ofertą dla użytkowników Amigi, daje pojemność pamięci wystarczającą do większości zastosowań naszego komputera. Jednakże opisane usterki stawiają pod znakiem zapytania użyteczność rozszerzenia.

HIGHTOWER

ZALETY:
+ łatwość montażu.

WADY:
- konieczność rozkręcenia komputera podczas montażu, a co za tym idzie, utrata gwarancji na komputer,
- wadliwa współpraca z programami użytkowymi,
- niemożność zmiany pojemności rozszerzenia (pracuje ono tylko „w trybie" 2 MB),
- zła praca zegara.



AMOS cz. II

W poprzednim artykule przedstawiłem AMOSa tak, jakbym to czynił w kampanii reklamowej tego języka, a więc dość ogólnie. Tym razem przejdę do konkretów. Polecam ten artykuł zwłaszcza tym, którzy mają nieco inne ambicje, niż łamanie joysticka i kopiowanie dysków za pomocą X-Copy.

Bardzo często słyszy się pojęcia: język wysokiego lub język niskiego poziomu. Otóż językiem niskiego poziomu jest np. asembler. Komputer wykonuje program w asemblerze bezpośrednio — jeden rozkaz odpowiada jednemu rozkazowi procesora. Asembler to w zasadzie elementarny język maszyny, rozkaz jest liczbą, którą procesor odczytuje bezpośrednio.

Inną budowę mają języki wysokiego poziomu. Jedna instrukcja tego języka to szereg nawet kilkudziesięciu rozkazów asemblera. Ale dlaczego? Dlatego, aby ułatwić pracę piszącemu program. Jeżeli napiszemy w AMOSie:

Print „AMOS"

to komputer sprawdzi poprawność wpisanej instrukcji, rodzaj wyprowadzanej na ekran zmiennej lub stałej, odnajdzie w pamięci adres pod którym znajduje się tekst: „AMOS", itd. Źle napisana instrukcja w AMOSie, czy też w innym języku wysokiego poziomu wywoła tylko komunikat o błędzie. Natomiast błąd w asemblerze w dziewięciu przypadkach na dziesięć powoduje GURU...

Nie ma niestety róży bez kolców. Program w języku wysokiego poziomu jest dużo, dużo wolniejszy od programu maszynowego. Do tego jeszcze dochodzi fakt, że nie daje on połowy tych możliwości, które ma język niskiego poziomu. AMOS to jednak coś naprawdę wyjątkowego — ma on liczne instrukcje żywcem wyjęte z asemblera. W zasadzie trudno wymyśleć coś, co dałoby się zrobić w asemblerze, a nie udawałoby się w AMOSie.

Podsumowując:
- komputer wykonuje program odczytując liczby z komórek pamięci (tzw. kod maszynowy);
- asembler wymyślono po to, aby liczby w komórkach zastąpić słowami, które łatwiej jest zapamiętać;
- asembler jest językiem niskiego poziomu;
- jedna instrukcja języka wysokiego poziomu to kilkadziesiąt rozkazów kodu maszynowgo;
- język wysokiego poziomu jest bardzo wygodny, ale powolny i nie daje takich możliwości jak np. asembler;
- wyjątkiem jest AMOS, bardzo szybki, wygodny i dający olbrzymie pole do popisu zarówno dla początkujących, jak i dla zaawansowanych.

To tyle teorii, a teraz przechodzimy już do samego AMOSa. Po wczytaniu na ekranie powinien pokazać się prostokąt z nazwą programu, informacją o jego autorze, firmie itd. Gdy naciśniemy dowolny klawisz, ukaże się edytor z migającym kursorem, gotowy do wpisywania tekstu programu. Możemy teraz wpisać jakiś program lub wczytać go z dysku. Wczytajmy więc cokolwiek. Klawiszami Shift+F1 uruchamiamy opcję „Load", co oznacza po prostu „ładuj".

Gdy zakończymy operację czytania z dysku, to w edytorze powinien pojawić się tekst wczytanego programu (wybierajmy tylko te, które kończą się na „AMOS"). Wśród tego bełkotu dziwnych słów występuje sporo takich, które zaczynają się od dużej litery a następnie pisane są małymi. Są to właśnie instrukcje AMOSa. Jedna instrukcja w AMOSie składa się z jednego lub więcej wyrazów, z których każdy spełnia powyższy warunek pod względem budowy.

Instrukcja powoduje wykonanie jakiejś operacji przez komputer. Często potrzebuje ona jednak jeszcze dodatkowej informacji precyzującej zakres jej działania. Taką funkcję spełniają parametry, które umieszcza się po tekście instrukcji. Mogą one mieć rozmaity wygląd: liczby, wyrazy pisane tylko dużymi literami, bądź też np. dowolnej budowy teksty umieszczone w cudzysłowie.

Weźmy np. instrukcję wykonującą operację rysowania kreski na ekranie. Wiadomo, że kreska taka może być pozioma, pionowa, skośna, krótka, długa itd. Do sprecyzowania wyglądu tej kreski służą właśnie parametry. Zobaczmy, jak to wygląda:

Draw To — instrukcja służąca do narysowania kreski.
X1,Y1 — współrzędne punktu na ekranie, w którym ta kreska ma się zaczynać,
X2,Y2 — współrzędne punktu na ekranie, w którym kreska ma się kończyć.

A całość zapisujemy tak:

Draw X1,Y1 To X2,Y2

Wpiszmy więc dla przykładu:

Draw 0,0 To 100, 100

i naciśnijmy klawisz F1, co spowoduje uruchomienie programu. Działa? Musi. Dodam tylko, że oczywiście nie trzeba pisać instrukcji zaczynając od dużej litery i kontynuując małymi. Gdy naciśniemy klawisz Return na końcu linii, kursor przejdzie do następnej linii, a wyrazy kryjące pod sobą nazwy instrukcji zamienią się automatycznie na odpowiedni format. Gdy tak się nie stanie, znaczy to, że wpisaliśmy coś nieprawidłowo.

Wracając do wpisanego programu: powoduje on narysowanie kreski od punktu o współrzędnych 0,0 to jest w lewym górnym rogu ekranu do punktu o współrzędnych 100,100. Ekran to jakby układ współrzędnych rysowany w szkolnym zeszycie z tą tylko różnicą, że oś Y rośnie w dół.

Tak więc podstawowymi elementami AMOSa (a także innych języków wysokiego poziomu) są instrukcje. Aby sprecyzować ich działanie, potrzebne są parametry. Parametry te to albo tekst, albo liczba. Konieczna jest jednak także możliwość zadefiniowania zmiennej (tak jak w matematyce), która przechowywać będzie jakąś wartość lub tekst. Zmienna taka, wstawiona w odpowiednie miejsce, musi tę wartość reprezentować. Powinna także dać się modyfikować. Po co? Przypuśćmy, że mamy zrobić program pobierający od użytkownika liczbę (będącą np. rokiem jego urodzenia) i sprawdzający, czy jest to rok przestępny. O wyniku powinniśmy użytkownika poinformować, aby wynagrodzić mu trud wklepywania swoich osobistych personaliów. Jak więc to zrobić? Musimy przecież pobrać od niego czterocyfrową liczbę i wykonać na niej jakieś działania. M.in. do tego celu służą zmienne.

Za pomocą odpowiedniej instrukcji pobieramy od użytkownika rok i przypisujemy go pewnej zmiennej. Odtąd zmienna ta, użyta gdziekolwiek w dalszej części programu, reprezentować będzie wartość odpowiadającą wprowadzonej dacie urodzin (za wyjątkiem przypadku, gdy zostanie ona zmodyfikowana). Teraz powinniśmy jakoś sprawdzić, czy dany rok jest przestępny. Tu wystarczy tylko sprawdzić podzielność przez cztery. Dokonujemy więc operacji uzyskania reszty z



dzielenia wprowadzonej liczby przez cztery. Resztę tę możemy wpisać do tej samej zmiennej, która reprezentowała rok, gdyż ten nie będzie nam już potrzebny. Następnie należy się pochwalić wpatrzonemu w monitor człowiekowi, czego dokonaliśmy. Jeżeli nasza zmienna ma wartość zerową (pamiętajmy cały czas, że jest to już teraz reszta z dzielenia wartości roku przez cztery), to rok urodzenia jest rokiem przestępnym, jeżeli nie — rokiem zwykłym. Oto cała filozofia. Myślę, że chyba już każdy z Was dostrzega, jak ważne są zmienne.

Zmienne AMOSa można podzielić na trzy typy. Pierwszy z nich (kolejność jest oczywiście umowna) to ZMIENNE CAŁKOWITE, mogące przechowywać wartości liczb całkowitych z zakresu od −147483648 do 147483648. Zapisujemy je w programie dowolnym ciągiem znów (byle nie rozpoczynającym się od nazwy instrukcji AMOSa) nie zawierającym spacji, rozpoczynającym się od litery. Gdy zmienna jest napisana poprawnie, AMOS zamieni ją na duże litery. Wszelkie odstępstwa od tej reguły to błąd. Oto kilka prawidłowych przykładów: ZMIENNA, ZMIENNA2, ROK_URODZENIA itd. Nieprawidłowo zdefiniowane zmienne to np.: 2 ZMIENNA, ZMIENNA 2, a także np. DOM — ponieważ „Do" jest instrukcją AMOSa. Jeśli przejdziemy do następnej linii, to zmienna „DOM" automatycznie zostanie rozdzielona na „Do M", co pozwoli zorientować się w błędzie.

Drugi typ zmiennych — ZMIENNE RZECZYWISTE — mogą przechowywać liczby rzeczywiste (całości i ułamki) z zakresu od 1E-19 do 1E+18. Zapisujemy je na takich samych zasadach jak zmienne całkowite, ale kończymy znakiem #, np.: ZMIENNA#, ZMIENNA2#, ULAMEK# itd. W obliczeniach zachowują one dokładność do siedmiu cyfr znaczących. Specjalną zmienną rzeczywistą jest PI# przechowująca wartość liczby pi.

Nasuwa się pytanie, po co nam zmienne całkowite, skoro zmienna rzeczywista może przechowywać także całkowite wartości. Tak, ale obliczenia dokonywane na zmiennych całkowitych są dużo szybsze i w miarę możliwości proponuję z nich korzystać.

Ostatni typ zmiennych to ZMIENNE TEKSTOWE. Tworzymy je na takich samych zasadach jak zmienne całkowite, z tym, że na końcu dodajemy znak $, np.: ZMIENNA$, ZMIENNA2$, IMIE$. Mogą one przechowywać łańcuchy dowolnych znaków (tekst) o długości do 65500 znaków.

Zatem wiemy już, co to są zmienne i jakie mamy ich typy. Teraz poznajmy jeszcze, jak nadawać im wartości (zmienne całkowite i rzeczywiste mają początkowo wartość 0, a zmienne tekstowe reprezentują pusty łańcuch znaków). Przyporządkowania takiego dokonujemy za pomocą znaku równości, np.:

ZMIENNA=10
ZMIENNA#=10.1546

i jeżeli zamiast liczby użyjemy nazwy ZMIENNA, to komputer podczas wykonywania programu wstawi wartość 10, a w przypadku nazwy ZMIENNA# — 10.1546. Należy także dodać, że przypisanie zmiennej całkowitej liczby rzeczywistej nie spowoduje błędu, lecz zignorowanie części ułamkowej liczby, np.:

ZMIENNA=3.14159 (zmienna ZMIENNA przyjmie wartość 3).

Tekst przypisywany zmiennej tekstowej należy ująć w cudzysłowy, np.:

ZMIENNA$=„AMOS Basic".

Pamiętajmy, że zmienna, do której chcemy coś przypisać, znajduje się zawsze po lewej stronie znaku równości, a wartość jej przypisywana po prawej. Próby odwrócenia kolejności, np. 10=ZMIENNA, spowodują błąd.

Mówię o tym, ponieważ swego czasu, gdy zaczynałem swoją przygodę z komputerami, bardzo mnie to dziwiło. Z matematycznego punktu widzenia odwrotne napisanie równości nic przecież nie zmienia. Jeszcze bardziej zszokował mnie zapis:

ZMIENNA=ZMIENNA+10

W matematyce coś takiego jest absolutnie niedopuszczalne (czysta sprzeczność), natomiast w AMOSie (i wielu innych językach, np. w BASIC) powoduje po prostu zwiększenie zmiennej ZMIENNA o wartość 10.

Jeśli już coś do czegoś dodaliśmy, to poznajmy dostępne w AMOSie operacje arytmetyczne wg kolejności ich wykonywania:

^ — potęgowanie
* / — mnożenie i dzielenie
MOD — reszta z dzielenia
\+ − — dodawanie i odejmowanie
AND — logiczne „i" (koniunkcja)
OR — logiczne „lub" (alternatywa)
NOT — logiczne „nie" (zaprzeczenie)

i bardziej skomplikowane:

Inc ZMIENNA — powoduje zwiększenie zmiennej ZMIENNA o 1
Dec ZMIENNA — powoduje zmniejszenie zmiennej ZMIENNA o 1.
Add ZMIENNA, O_ILE,MIN To MAX — do zmiennej ZMIENNA dodana jest wartość zmiennej O_ILE. Jeżeli w wyniku tej operacji ZMIENNA przekroczyła wartość MAX, to zostanie jej nadana wartość MIN. Jeżeli ZMIENNA przyjęła wartość mniejszą od MIN, to zostanie jej nadana wartość MAX.

Wszystkie powyższe operacje mogą dotyczyć liczb, zmiennych całkowitych i rzeczywistych. Natomiast w przypadku zmiennych tekstowych dopuszczalne są jedynie operacje dodawania i odejmowania.

Chciałbym teraz zapoznać szanownego czytelnika z kilkoma instrukcjami, które będą pomocne w pisaniu prostych przykładów ilustracyjnych nasze rozważania.

Print

Instrukcja służąca do wyprowadzania na ekran liczb, tekstów i wszelkich zmiennych. Za pomocą jednej instrukcji „Print" możemy ich wyprowadzać wiele, z tym, że muszą być one rozdzielone przecinkiem lub średnikiem. Średnik spowoduje wyprowadzenie kolejnych liczb, tekstów, lub zmiennych jedna za drugą, obok siebie, natomiast przecinek przesunie każdą z nich do najbliższej następnej kolumny tabulacji. Oto przykład:

Print 1234, „To jest tekst.", ZMIENNA, ZMIENNA$

lub też

Print 1234; „To jest tekst."; ZMIENNA; ZMIENNA$

Input

Instrukcja, która jest jakby przeciwieństwem poprzedniej. Czyta ona z ekranu to co napiszemy, aż do chwili naciśnięcia klawisza Return, a następnie wpisuje to do zmiennej, którą podajemy jako parametr tej instrukcji. Może ona zawierać także komentarz dotyczący wprowadzonej wartości, np.:

Input ROK

lub też

Input „Wprowadź rok swojego urodzenia:"; ROK

If... Then...

Jedna z tzw. instrukcji warunkowych. Jeżeli spełniony jest warunek, który występuje zaraz po „If", to wykonywane są instrukcje znajdujące się po „Then", np.:

if RESZTA=0 Then Print „Urodziłeś się w roku przestępnym"

Instrukcja ta ma jednak wadę: wszystkie instrukcje, które mają się wykonać w wyniku spełnienia warunku, muszą znajdować się w jednej linii programu. Często lepszą, choć bardziej rozbudowaną formą tej instrukcji jest:

```
If warunek
Tu znajdują się instrukcje,
które są wykonywane, gdy warunek jest spełniony
Else
Tu znajdują się instrukcje,
które są wykonywane, gdy warunek
nie jest spełniony
End If
```

Należy oczywiście dodać, że cały człon zawierający słowo Else oraz instrukcje po nim występujące, jest opcjonalny, tzn. można go pominąć. A teraz przykłady:

```
If RESZTA=0
  Print „Urodziłeś się w roku przestępnym"
End If
```

albo

```
If RESZTA = 0
  Print „Urodziłeś się w roku przestępnym"
```



```
Else
  Print „Urodziłeś się w roku zwykłym"
End if
```

A teraz program, który chcieliśmy napisać na początku artykułu:

```
Input „Wprowadź swój rok urodzenia:"; ROK
RESZTA=ROK mod 4
If RESZTA=0
  Print „Urodziłeś się w roku przestępnym"
Else
  Print „Urodziłeś się w roku zwykłym"
End if
```

Po wpisaniu możemy go uruchomić za pomocą klawisza F1. Po zakończeniu wykonywania programu należy nacisnąć spację, aby wrócić do edytora. Chciałbym przedstawić jeszcze jeden prosty przykład — program liczący równanie kwadratowe. Proszę zwrócić uwagę na operacje na zmiennych i komputerowy sposób podejścia do problemu. Nowością jest tu instrukcja: **Sqr (ZMIENNA)** obliczająca pierwiastek kwadratowy ze zmiennej ZMIENNA. Rozszyfrowuje także znaki występujące przy porównaniach:

= równość (wiadomo)
> mniejszy od ...
> większy od ...
<> nierówność
>= równy lub większy od ...
<= równy lub mniejszy od ...

A teraz program:

```
Print „Program liczy pierwiastki równania kwadratowego"
Print „postaci: A*X2+B*X+C=0"
Print „Podaj współczynniki A,B,C:"
Input „A=";A
Input „B=";B
Input „C=";C
DELTA=B^2-4*A*C
If DELTA>=0
Print „A oto wynik:"
  If DELTA>0
  X1=(-B-Sqr (DELTA)) / (2*A)
  X2=(-B+Sqr (DELTA)) / (2*A)
  Print „X1=";X1
  Print „X2=";X2
Else
  X=-B/(2*A)
  Print „X=";X
End If
Else
  Print „Delta jest mniejsza od zera."
  Print „Równanie nie ma rozwiązań."
End If
```

Zwróćmy uwagę, że w tym programie wewnątrz jednej instrukcji warunkowej znajduje się druga. Oto algorytm:

Jeśli DELTA jest większa lub równa zero, to
  jeśli jest większa od zera, to
  równanie ma dwa pierwiastki
  i trzeba to pokazać
a jeśli nie jest większa od zera, tylko równa zero, to
  równanie ma jeden pierwiastek
  i trzeba to pokazać
(koniec wewnętrznej instrukcji warunkowej)
a jeśli DELTA nie jest większa lub równa zero, to
  poinformuj użytkownika, że niech sobie sam wyznacza
  pierwiastki, równania którego delta jest mniejsza od zera
(koniec zewnętrznej instrukcji warunkowej)

Myślę, że tak obrazowe wyjaśnienie zrozumie każdy. Jeśli by tak jednak nie było, to proszę o listy lub telefony. Życzę miłej zabawy.

RAFAŁ BORZYŃSKI (RABOCOST)

POLAK POTRAFI

Jakiś czas temu na naszym rynku zaczęły pojawiać się wytwory rodzimych programistów. Polscy twórcy wszelakiego osprzętu dla Amigi nie pozostali oczywiście w tyle i ich wysiłek owocuje wieloma ciekawymi przystawkami.

Ostatnio na rynku rozszerzeń pamięci dla Amigi 500(+) zagościła warszawska firma Elsat, której dziełem jest nowe, bardzo ciekawe rozszerzenie pamięci. W odróżnieniu od setek innych dostępnych na rynku rozszerzeń dla Amigi 500(+) ma ono wiele nowych możliwości. Przede wszystkim umożliwia rozbudowę pamięci aż o 8 MB. Wykorzystywane jest przy tym złącze procesora, a nie, jak w przypadku większości pozostałych rozszerzeń, złącze rozszerzenia pamięci pod klawiaturą. Takie rozwiązanie pociąga za sobą szereg konsekwencji, w większości pozytywnej natury. Tak więc możliwe jest podłączenie aż ośmiu megabajtów pamięci, która jest rozpoznawana przez Amigę jako FAST, a nie SLOW lub CHIP. Trzeba pamiętać, że dopiero w FAST procesor może osiągnąć pełną szybkość, co w praktyce ma takie znaczenie, że Amiga wyposażona w pamięć FAST pracuje około 30% niż jej koleżanka dysponująca SLOW lub CHIP.

Ważny jest też nieskomplikowany montaż — dzięki temu, że rozszerzenie korzysta ze złącza procesora, nie potrzeba koniecznie otwierać Amigi (i tracić gwarancję), jak też nie trzeba grzebać się w jej wnętrzu (na przykład wkładać podstawki pod Gary'ego). Warto tu dodać, że rozszerzenie jest „przelotowe", co oznacza, że nie blokuje sobą złącza procesora i nie pozbawia nas możliwości dalszej rozbudowy.

Dla mniej ambitnych użytkowników Amigi, których narzędziem „pracy" jest częściej joystick niż myszka, przeznaczony jest z kolei wynalazek krakowskiej firmy Mapasoft. Jest to malutka przystawka wtykana do gniazda myszki, umożliwiająca jednoczesne podłączenie do tego gniazda i joysticka, i myszki. Nie jest to jednak prosty rozgałęziacz — malutki układ wewnątrz przystawki rozpoznaje, które z urządzeń jest aktualnie używane i tylko jego sygnały przesyła do Amigi. Z pewnością niejeden namiętny gracz przestanie niebawem narzekać na niszczenie gniazda bezsensownym przełączaniem.

Testy omawianych tu urządzeń zamieścimy niebawem w C&A.

ANDY

ENGLISH TEACHER

ENGLISH TEACHER jest nieźle przemyślanym, rozbudowanym programem uczącym języka angielskiego. Już na pierwszy rzut oka dostrzegamy staranność wykonania grafiki (zastosowano tzw. „3D look" znany doskonale posiadaczom Amig wyposażonych w Kickstart 2.0), czytelnie umieszczone wskaźniki i przejrzysty ekran roboczy dopełniają reszty. Również pomysłowa oprawa dźwiękowa i synteza mowy udanie harmonizuje z całością. Program przedstawiono jako nauczyciela, dysponującego na tyle rozbudowaną wiedzą, by usatysfakcjonować każdego. Niemożliwe? — przyjrzyjmy się.

DZIAŁANIE

Program na bieżąco ocenia nasze postępy w nauce (informują nas o nich słoneczko o wielu obliczach i wskaźnik w górnym prawym rogu ekranu), a także komentuje wyniki nauczania i wskazuje poprawne odpowiedzi. Warto w tym miejscu dodać, że owe uwagi potrafią co poniektórych sprowadzić na ziemię. A gdy czas upłynie... na ekranie pojawia się latający szczur — i tu uwaga dla myśliwych — jest ustrzeliwalny. Po rozwiązaniu danego działu program wystawia ocenę (za czwórkę, piątkę i szóstkę przewidziano nagrody).

Przejdźmy do konkretów. Program ma strukturę modułową i dzieli się na trzy główne części: NAUCZANIE, TESTY i KRZYŻÓWKI.

1. Nauczanie

Nauczanie jest wieloetapowe i usystematyzowane, godne pochwały jest również staranne i fachowe dobranie materiału, z którego się uczymy. Pytania są ponawiane aż do momentu, kiedy odpowie się prawidłowo. W module zostały wyodrębnione następujące opcje:

układanie pytań — zadaniem uczącego się jest ułożenie pytań do zdań ułożonych przez komputer. Ważne jest to, że zdań jest dużo, zaś dla każdego z nich przewidziano po kilka odpowiedzi, co gwarantuje pewną „inteligencję" komputera przy interpretowaniu odpowiedzi „ucznia".

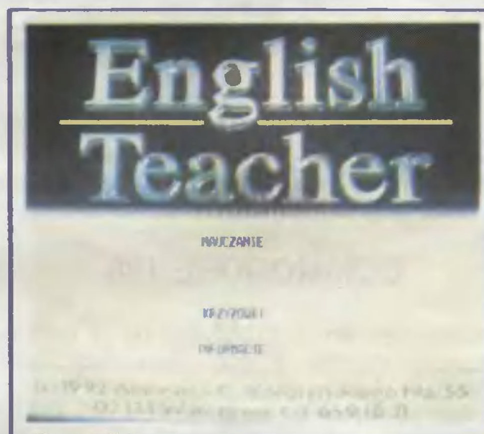
tłumaczenie zdań — zadaniem uczącego się jest przetłumaczenie z języka polskiego na angielski zdań dyktowanych przez komputer. Tak jak w opcji „układanie pytań" przewidziano po kilka odpowiedzi.

uzupełnianie zdań — w tym etapie uczący się ma za zadanie wstawić odpowiedni zwrot w wyróżnione miejsce w zdaniu (do wyboru jest pięć możliwości), w razie pomyłki program zawsze podaje poprawną gramatycznie, uzasadnioną odpowiedź.

dokańczanie zdań — należy w tym miejscu wstawić dalszy, logiczny, poprawny gramatycznie ciąg zdania, którego początek podaje nam komputer.

słownictwo — wyodrębniono tu kilka grup wyrazów m.in.: przymiotniki, zwierzęta, ciało, komputer, jedzenie, dom, czasowniki z nieregularną odmianą (1), czasowniki z nieregularną odmianą (2), natura, osoba, czas, podróż, ubranie itd. I tu podczas nauki spotyka nas miła niespodzianka: program rozróżnia wiele synonimów danego wyrazu (również je wyświetlając i wymawiając).

W przypadku pierwszych czterech opcji mamy też do wyboru po kilka poziomów trudności (najczęściej od 0 do 6). Wraz ze wzrostem poziomu trudności komputer zaczyna posługiwać się różnymi czasami, konstrukcja staje się bardziej skomplikowana i wyrafinowana, łącznie z użyciem idiomów i nietypowych konstrukcji. Dzięki takiej koncepcji z programu może korzystać niemal każdy, niezależnie od poziomu wiedzy z języka angielskiego.



PRODUCENT I DYSTRYBUTOR: Alderan S.C., ul. Korotyńskiego 19a/55, 02-123 Warszawa, tel. (02) 659-18-21

2. Testy

Ta część programu jest nieco podobna do „Nauczania", z pewnymi jednak różnicami, m.in. wprowadzono tu punktację i limit czasowy — jak na prawdziwy egzamin przystało. Oto opcje:

układanie zdań — komputer wyświetla zdanie, zaś uczący się ma za zadanie skonstruować stosowne pytanie; podobnie jak w części „Nauczanie", również i tutaj komputer rozpoznaje wiele poprawnych form.

słownictwo — tym razem zadanie jest „banalne", należy przetłumaczyć dyktowane przez komputer zwroty i wyrażenia.

tłumaczenie zdań, uzupełnianie zdań — podobnie jak w dziale „Nauczanie", lecz z innymi kryteriami oceny, pozwalającymi określić postępy w nauce.

dokańczanie zdań — tutaj uczący się jest testowany ze znajomości gramatyki angielskiej.

3. Krzyżówki

Dostępnych jest dziesięć krzyżówek, każda o innej tematyce, każda w starannej, ładnej oprawie graficznej. Brawa należą się też za dobrze przemyślaną obsługę i sterowanie. Po przytrzymaniu RETURN program wskazuje błędy i literówki, które popełniliśmy.

Krzyżówki pogrupowano m.in. według kluczy:
- nazwy geograficzne,
- imiona i ich zdrobnienia,
- przymiotniki.

Po rozwiązaniu każdej krzyżówki komputer podaje hasło, gdy zaś zgromadzi się już wszystkie dziesięć, można przesłać je na adres firmy Alderan, dzięki czemu weźmie się udział w losowaniu wielu (podobno cennych!) nagród.

PODSUMOWANIE

Program English Teacher można chyba polecić każdemu, kto ma cokolwiek wspólnego (lub zamierza mieć) z językiem angielskim. Program zawiera szereg oryginalnych pomysłów i rozwiązań, również jego wykonanie jest bez zastrzeżeń. Co ważne — NIE JEST NUDNY I MONOTONNY! Pozwala bez większych „cierpień" opanować spory materiał językowy i utrwalić go w pamięci, „doszlifować" gramatykę, poznać niuanse językowe. Dodatkowo po pomyślnym przejściu każdego modułu czeka na ucznia niespodzianka, a mianowicie trzy w pełni dopracowane i atrakcyjne gry: Pacman, Puzzle i Tetris.

English Teacher'a można bez ryzyka nazwać programem uniwersalnym. Decyduje o tym głównie wielopoziomowe nauczanie (dzięki któremu użytkownik może dostosować program do swoich potrzeb i stopnia znajomości języka) i rozbudowany, bardzo dobrze opracowany materiał, podzielony tematycznie i gramatycznie. Należy w tym miejscu zaznaczyć, że gramatyka jest zawsze przedstawiona w praktycznym zastosowaniu, nie jako sucha teoria, czy oderwana od rzeczywistości regułka. Nie bez znaczenia jest uatrakcyjnienie programu, widać iż autorzy położyli duży nacisk w tym kierunku — począwszy od menu programu i środowiska poprzez efekty specjalne (świetne latające szczury!), a skończywszy na bonusowych grach — nagrodach. W żadnym momencie nauki nie odczuwamy typowego (znanego każdemu ze szkoły) znużenia i idącego w parze z nim zobojętnienia. Jest też jeszcze jedna zaleta Nauczyciela Angielskiego — możemy go w każdej chwili wyłączyć i pójść, dajmy na to, na piwo, co gwarantuje nam komfort psychiczny, jakże ważny przy nauce.

Wierzę, że spory wysiłek programistów (program został napisany w całości języku maszynowym, dzięki czemu jest szybki i zajmuje niewiele pamięci) przyniesie korzyść wszystkim — pragnącym uczyć się, uczącym, przypominającym i utrwalającym sobie język angielski. English Teacher jest godnym następcą starszego produktu firmy ALDERAN, programu Word Teacher, i to nie tylko ze względu na lepsze wykonanie, ale przede wszystkim na to, że umożliwia on, oprócz nauki słówek, także szersze poznanie języka angielskiego i jego gramatyki.

TOMASZ KOWALSKI

KĄCIK POCZĄTKUJĄCEGO



Cykl ten przeznaczony jest co prawda dla początkujących użytkowników komputerów C-64/128, aliści nie znaczy to, że bardziej zaawansowani nie znajdą tu czegoś dla siebie.

ZACZYNAMY

Jeśli już w jakiś sposób wszedłeś w posiadanie C-64 lub C-128 powinieneś pamiętać, że komputer ma określoną wytrzymałość, a więc jest urządzeniem delikatnym i wymagającym specjalnego traktowania. Oto kilka rad wchodzących w zakres komputerowego BHP:

1. Nigdy nie podłączaj i nie odłączaj żadnych urządzeń przy włączonym komputerze. Układy scalone C-64/128 są bardzo wrażliwe na nagromadzone ładunki elektrostatyczne. Iskra, powstała wskutek podłączenia np. modułu do działającego komputera może być przyczyną awarii. Popularne wśród posiadaczy Commodore jest przekładanie joysticka z jednego portu do drugiego przy włączonym komputerze (np. wtedy, gdy dana gra „chodzi na drugim porcie". Takie wygodnictwo może Cię kosztować dwukrotnie więcej, niż mógłbyś wydać na drugi joystick. Uwaga ta dotyczy szczególnie wersji C-64C i C-64 VGS, których najwięcej można uświadczyć w serwisach, czekających na naprawę.

2. Jeżeli się nie znasz, nie otwieraj i nie próbuj sam naprawić komputera. Nawet w przypadku prostej wymiany bezpiecznika zwróć się po pomoc do kogoś obeznanego z budową komputera. Często los jest tak złośliwy, że z małego uszkodzenia robi się wielka awaria, oczywiście z naszą wydatną pomocą.

3. Zachowaj kolejność przy uruchamianiu komputera i urządzeń z nim współpracujących. Należy włączać najpierw monitor oraz inne urządzenia zewnętrzne, a dopiero na końcu komputer. Przy wyłączaniu postępuj w odwrotnej kolejności.

4. Jak najrzadziej korzystaj z przełącznika sieciowego komputera. Częste „pstrykanie" nim (włączanie/wyłączanie) może być przyczyną awarii komputera. Po wyłączeniu C-64 należy odczekać kilka sekund i dopiero potem włączyć komputer.

5. Dbaj o wzrok, masz tylko jedne oczy. Pamiętaj o zachowaniu odpowiedniej odległości od monitora lub telewizora. W miarę możliwości stosuj filtry ekranowe. Rób przerwy w pracy z komputerem.

Po tej solidnej porcji rad i przestróg, warto się dowiedzieć, jakie w ogóle typy komputerów C-64/128 są dostępne oraz jakie urządzenia zewnętrzne (drukarki, monitory, stacje dysków itp.) najlepiej do nich podłączyć.

COMMODORE 128

Komputer ten produkowany był w dwóch wersjach: zwykłej i z wbudowaną stacją dysków 1571 (128D). Do obu tych modeli można podłączyć firmową stację dysków VC 1571, posiadającą dwie głowice, a więc możliwość dwustronnego zapisu dyskietek. Do obu typów komuterów można podłączyć magnetofon, np. Datasette 1530, TurboCorder, posiadający dziesięciostykowe złącze krawędziowe. C-128 (D) może pracować w trzech trybach: emulacji C-64 (zgodny z Commodore 64), C-128 (ze wspaniałym BASIC i 128 kB pamięci) i CP/M. Powstało jednak bardzo niewiele programów działających w trybie 128. Tak więc w większości przypadków komputer ten wykorzystywany jest jako zwykła „sześćdziesiątkaczwórka".

Obok standardowych drukarek typu MPS 801/803 lub innych posiadających interfejs szeregowy SERIAL, do C-128 można podłączyć w zasadzie każdą drukarkę wykorzystując port użytkownika (USER PORT) lub stosując specjalny interfejs pozwalający podłączyć ją do gniazda SERIAL. Do komputera można podłączyć telewizor dzięki wyjściu RF. Wśród najczęściej używanych monitorów spotyka się zielone monitory polskiej produkcji NEPTUN 156. Ponieważ C-128 posiada także cyfrowe wyjście RGB, pozwalające uzyskać 80 znaków w linii (tylko tryb 128 i CP/M), dobrym wyjściem jest zakup monitora kolorowego (np. 1084S) posiadającego takie złącze. C-128 posiada również gniazdo MEMORY EXPANSION PORT, do którego można podłączyć rozszerzenie pamięci lub dowolny moduł.

COMMODORE 64

Komputer ten od roku 1982, czyli od momentu powstania cieszy się olbrzymią popularnością. Produkowanych jest kilka typów C-64. Pierwszy to stary typ C-64, czyli taki, od jakiego zaczęto. Szarobrunatna obudowa o obłych kształtach (złośliwi atarowcy porównywali go z podkładem kolejowym). Po pewnym czasie firma Commodore postanowiła unowocześnić linię C-64 i zmieniono obudowę na białą, o nowoczesnych kształtach (troszkę podobna do C-128). W ten sposób powstało C-64II. Równolegle z nim, głównie dla potrzeb amerykańskiego rynku powstała ośmiobitowa wersja laptopa — SX64. Posiadał on wbudowany kolorowy monitor 5", stację dysków 1541. Był w pełni kompatybilny z C-64. Jego podstawową wadą był brak gniazda magnetofonu. Po wyprodukowaniu niewielkiej ilości SXów, firma Commodore zrezygnowała z tego modelu (z powodu nieopłacalności produkcji) i postanowiła zająć się ulepszeniem starego C-64. Ulepszenie polegało głównie na zmodernizowaniu zasilacza i układu dźwiękowego SID oraz dodaniu literki C do nazwy (Commodore 64C). Unowocześnienie SID okazało się niestety niewypałem (mimo, że był tańszy w produkcji); tzw. samplingi są wręcz nieprzyzwoicie wytłumiane, wskutek czego muzyka na C-64, oparta obecnie w pewnej mierze na dźwiękach digitalizowanych, jest o wiele uboższa.

Wreszcie, jako najnowsze 8-bitowe dziecko firmy Commodore ukazał się, wzorem firmy Atari, zestaw komputerowy przeznaczony głównie do gier. Zestaw taki zyskał nazwę Commodore 64 Video Game System (VGS). W skład kompletu wchodził komputer (C-64C w obudowie starego typu, w jasnych kolorach), joystick oraz cartridge z trzema grami. Podstawową wadą tego komputera, obok nowego układu SID, jest zmieniony procesor graficzny VIC, przez co pewien odsetek pro-

gramów (głównie użytkowych) nie działa poprawnie.

Do każdego z wyżej wymienionych rodzajów Commodore 64 można podłączyć magnetofon posiadający złącze krawędziowe (Datasette 1530, TurboCorder itp.). Obecnie produkowane są dwa typy stacji dysków do tego komputera: VC 1541II oraz Turbo 9900 (test zamieszczony był w Bajtku). Rzadziej można spotkać nieprodukowane już, ale wciąż dobre stacje dysków VC 1541, charakteryzujące się masywną budową oraz wbudowanym zasilaczem.

C-64 może być podłączony do każdego monitora i telewizora, posiadającego wyjście AV. Do telewizora można C-64 podłączyć dodatkowo przez wyjście „antenowe" RF, dzięki wbudowanemu modulatorowi TV. Wszystkie drukarki wymienione przy C-128 mogą być także użyte przy współpracy z C-64.

OPROGRAMOWANIE

Omawiając komputery C-128 i C-64 nie można zapomnieć o długiej liście oprogramowania obejmującej gry i programy użytkowe. Nie będziemy omawiali gier, bo trzeba by poświęcić na to nie artykuł lecz kilka pokaźnych ksiąg. Wystarczy jedynie powiedzieć, że na świecie znajduje się obecnie ok. 12 tys. gier dla C-64 i 128 z czego ok. 7 tys. dostępnych jest w Polsce.

Z programów użytkowych należy wymienić graficzny system GEOS, którego najnowsza wersja dla C-128 i C-64 nosi numer 2.0.

Wśród najlepszych edytorów tekstu (z polskimi literami) dla C-64 znajdują się: GEOWRITE 2.1 (pracujący w środowisku systemu GEOS), POLSCRIPT, TEXTOMAT PLUS 64, STAR TEXTER 64, FONTMASTER II oraz edytor tekstu polskiej produkcji EDYTOR PL. Na C-128, obok edytorów dostępnych w trybie C-64, znajdują się dodatkowo: TEXTOMAT PLUS 128, STAR TEXTER 128, FONTMASTER 128, działające w trybie 80 znaków.

Z programów graficznych wymienić należy STAR PAINTER 64/128, GEOPAINT 64/128, AMICA PAINT, ART STUDIO, AFLI EDITOR. Bazy danych: SUPERCALC 64/128, DATA MANAGER 64/128, DATAMAT 128, DFILE 128, SUPERBASE 64, DBASE 2.0 CP/M (128).

Arkusze kalkulacyjne: SUPERCALC 64/ /128, SWIFTCALC 128, MULTIPLAN CP/M (128).

Najlepsze programy muzyczne to: MUSIC SYSTEM, MUSIC SHOP, ROMUZAK, VOICETRACKER, DRUMMAKER, ROCKMONITOR. Panuje wśród nich podział na monitory muzyczne (dźwięk zapisywany jest szesnastkowo) oraz edytory muzyczne, w których zapis dźwięku odbywa się klasyczną, nutową metodą.

To by było na tyle ogólnikowych informacji na temat komputerów C-64 i C-128. W następnych odcinkach zapoznamy Was z podstawowymi poleceniami języka BASIC obu tych komputerów.

cdn.

PIOTR LISZEWSKI

PALCEM PISANE
CZYLI
CO TO JEST I DO CZEGO SŁUŻY
TEXTOMAT PLUS 64/128

Kiedyś, gdy nie było mi jeszcze dane korzystać z dobrodziejstw techniki w postaci PC AT i jednego ze wspaniałych edytorów tekstu napisanych dla tego komputera, postanowiłem wycisnąć z małego Commodorka tyle, ile można w tej dziedzinie. Na brak oprogramowania nie mogłem narzekać, lecz możliwości, które oferowało, przyprawiało mnie o nerwowy zanik napięcia (nerwowego, rzecz jasna). Na pierwszy ogień poszedł FONTMASTER II, owszem, fajny, duży i po angielsku, lecz polskie literki zamiast bardzo potrzebnych znaczków "!", "#" czy "%" to lekkie przegięcie. Innym zaproponowanym mi edytorem był Polscript, polska mutacja programu EASYSCRIPT. Tu z kolei cierpiałem na chroniczny brak fontów, trudną obsługę i w ogóle brak komunikatywności. Moje wymagania po części spełnił PRINTFOX PL, który miał już to wszystko co chciałem, lecz niestety drukował „na dwa razy" jedną stronę i do tego były problemy z przeniesieniem pliku tekstowego na komputer klasy PC.

Wtedy to właśnie jako ostatniej deski ratunku uchwyciłem się programu TEXTOMAT PLUS. I to był strzał w dziesiątkę (nieźle to sobie wykombinowałem). Edytor ten posiada specjalny program TEXTOMAT ANPASSUNG pozwalający na tworzenie własnych czcionek, także polskich. Co jest dla mnie najważniejsze, polskie literki uzyskuje się przez naciśnięcie klawisza COMMODORE LOGO KEY i litery, do której ma być dorobiony „ogonek", np. aby uzyskać literę "ą" muszę, trzymając klawisz COMMODORE wcisnąć literkę "A". Ponadto program umożliwia pełną i szybką obróbkę tekstu w postaci dobrze widocznej na ekranie (na C-128 w trybie 80 znaków), druk w kilku krojach, pogrubianie, podkreślanie itp. czyli prawie wszystko to, co normalny edytor powinien potrafić. Niestety nie ma róży bez kolców. TEXTOMAT PLUS został napisany przez niemiecką firmę DATA BECKER, która, jak można się łatwo domyślić, zaopatrzyła swój edytor w niemieckie komentarze. Ponieważ język ten pozostaje dla mnie nadal czarną magią, miałem spore kłopoty z opanowaniem programu. Lecz teraz, po krwawych bojach, kiedy czuję się już cięższy o kilka kilogramów wiedzy na temat tego edytora tekstu, mogę się podzielić z Wami moimi wiadomościami. No to zaczynamy.

Na Commodore 64 wczytujemy program komendą LOAD"*",8,1. Na C-128 musimy wpisać BOOT"LADER".

Po pokazaniu się winiety początkowej komputer pyta się o znaki ekranowe [bildschirmzeichensatz] oraz znaki i parametry drukarki [druckerparameter]. Jeżeli wśród plików na dyskietce posiadacz zawierające polskie czcionki, to należy podać ich tytuły i potwierdzić naciśnięciem klawisza F1. Tytuł pliku ze znakami obrazu ma dopisaną literkę "b", natomiast plik znaków drukarki ma dopisaną literę "d".

Gdy komputer załaduje żądane znaki, pyta się o datę. Wpisujemy datę i przechodzimy dalej przez naciśnięcie F1. Następnie ekran zostanie wypełniony przez pole edycji. W górnym, lewym rogu, program podaje informacje na temat trybu, w jakim aktualnie jesteśmy, natomiast prawy, górny róg jest zajęty przez wskaźniki położenia kursora: pionowy i poziomy.

Całą linię dolną zajmuje MENU robocze, do którego można przejść naciskając F1 (wyjście — SHIFT+F1).

Opiszę teraz poszczególne opcje MENU.

EDIT — opcja ta umożliwia nam operację na plikach. Wczytywanie pliku [einlesen], wyszukanie i zamiana sekwencji znaków [suchen], skasowanie pliku [loeschen], operacje na blokach [block], łączenie tekstu z grafiką [grafik]. Dodatkowo w opcji BLOCK znajdują się podopcje: zaznaczanie bloku [mark], zaznaczanie końca tekstu [schieb], kopiowanie bloku [kopieren], kasacja bloku [loeschen], zapisanie bloku [speichern].

Funkcja **EINLESEN** umożliwia wczytanie pliku z dyskietki lub za pomocą modemu. Pliki tekstowe posiadają w tytule literkę "t".

FORMULAR — linia formatowa: [anlegen] tworzenie nowej linii formatowej (o tym powiem później), [einlesen] wczytanie gotowej linii formatowej (tytuł pliku z literką "f"), [loeschen] kasacja linii.

AUSGABE — czyli wyjścia: zapisanie tekstu na dyskietkę [speichern], podgląd tekstu [zeigen], drukowanie tekstu [druck], wysyłanie tekstu przez modem [senden], zapisanie pliku tekstowego w kodach ASCII [floppy].

DIENST — czyli różne opcje programu: [floppyoperationen] operacje na dyskietkach (formatowanie, weryfikacja, kasowanie itp.).

ANPASSUNG — służy do tworzenia nowych znaków i parametrów dla drukarki oraz ekranu. O programie TEXTOMAT ANPASSUNG napiszę już wkrótce; [bildschirmzeichensatz laden] wczytywanie nowych znaków ekranu, [programm beenden] koniec programu. [farbwahl] barwy ekranu.

TWORZENIE LINII FORMATOWEJ

Aby utworzyć nową linię formatową, należy wejść w opcję FORMULAR z MENU. Przez naciśnięcie klawisza F1 przechodzimy do podopcji ANLEGEN. Na ekranie zostaje wyświetlona pierwsza strona formularza. Kolejno od góry patrząc: [linker rand] — lewy margines; [rechter rand] — prawy margines; [oberer rand] — górny margines; [unterer rand] — dolny margines; [blattlaenge] — ilość linii na wydruku; [tabulatoren] — tabulacja; [zeichen pro zeile (bildschirm)] — ilość znaków w linii na ekranie; [uberschriften — fussnoten] — seryjne drukowanie danej sekwencji znaków w miejscu określonym przez użytkownika (np. numer strony gdzie automatyczną numerację stron uzyskujemy przez wstawienie znaczka "#"). Naciśnij F1 w celu przejścia do następnej strony formularza lub F2 w celu powrotu do MENU.

Po przejściu na drugą stronę od góry mamy: [zeichendichte] — wysokość znaku (tryb graficzny); [zeilenabstand] — odstęp między liniami; [blocksatz] — dołączanie bloków; [einzelblatt] — pojedyncza strona; [proportionalschrift] — pismo proporcjonalne; [wordwrap] — przenoszenie (dzielenie) słów; [anzahl laufwerke] — ilość działających stacji dysków (1 lub 2); [datum] — data. Poniżej znajduje się miejsce [floskeltasten] na stałe używane w korespondencji zwroty i wyrażenia, które zostają podpisane pod klawisze funkcyjne. Uzyskuje się je podczas edycji przez wciśnięcie klawisza CONTROL i wybranego klawisza funkcyjnego.

Naciskając ponownie klawisz F1 przechodzimy do ostatniej strony formularza, w której znajduje się funkcja zapisu na dyskietkę gotowej linii formatowej.

EDYCJA TEKSTU

Aby napisać tekst, musimy wyjść z MENU roboczego przez wciśnięcie SHIFT+F1. Jeżeli został wczytany plik zawierający polskie literki ekranowe, to najczęściej będzie się je uzyskiwało przez naciśnięcie COMMODORE LOGO KEY+literka. Jeżeli w matrycy ilość znaków na ekranie w linii została określona jako 40 (lub mniej), cały edytowany tekst będzie się przesuwał tylko w górę i w dół. Natomiast gdy liczba ta jest większa niż 40, to ekran będzie się przesuwał także w bok, co jest dość uciążliwe podczas pracy.

Koniec pisanego akapitu zaznacza się tzw. „karetką" czyli strzałką skierowaną w lewą stronę, a uzyskuje się ją przez naciśnięcie klawiszy SHIFT+ +RETURN. Wygląd (kształt) fontów nie jest widoczny na ekranie (już raz mówiłem o róży i jej kolcach).

Po ekranie poruszamy się za pomocą klawiszy kursora. Szybkie przejście na początek tekstu uzyskujemy przez wciśnięcie klawisza F5, natomiast przejście na koniec edytowanego tekstu — SHIFT+F5. Wstawianie liter lub zamianę można uzyskać wciskając klawisz INS.

W programie Textomat Plus podczas edycji używa się kodów sterujących, umieszczanych przy tekście, a pozwalających na np. centrowanie linii lub bloku, pogrubienie, justowanie itp. Linię kodów sterujących tworzymy po wciśnięciu kombinacji CONTROL+S (na ekranie pojawi się litera "S" w negatywie).

Oto spis kodów formatujących tekst:

bl — włączanie i wyłączanie funkcji BLOCK,
c — włączanie i wyłączanie centrowania. Centrowanie nie działa poprawnie z pismem pogrubionym i nie wpływa na rozmieszczenie rysunków,
lnxx — wstawienie xx spacji. xx = (00-99),
erxx — przesunięcie wydruku linii w lewo o xx spacji. Musi być umieszczony na początku linii,
lrxx — ustawia lewy margines o szerokości xx,
rrxxx— ustawia prawy margines o szerokości xxx. xxx = (000-240),
swxx — nastąpi zmiana strony jeżeli do jej końca pozostało mniej niż xx linii.

Kody dołączające tekst lub grafikę z dysku:

axx — występuje w miejscu, do którego zostanie wpisana zawartość pola danych o numerze xx. Działa tylko w trybie graficznym,
d — w miejscu wystąpienia tego kodu zostaje umieszczona data,
grx:nazwa — (x=0 — jedna stacja dysków) wczytuje z dyskietki blok graficzny o danej nazwie,
ndx:nazwa — do tekstu w pamięci zostaje dołączony tekst z dyskietki o danej nazwie.

Textomat plus zawiera dwadzieścia definiowanych kodów dla drukarki, możliwość ich stosowania zależy od posiadanej drukarki. Oto niektóre z nich:

abx — zmiana odstępu między wierszami (x=1,2,3),
dix — zmiana odstępu między literami w wierszu (x=10,12,15),
f — włącza i wyłącza pismo pogrubione (fettschrift). Dla drukarek MPS801-803 trzeba około dwukrotnie zmniejszyć szerokość linii.
o — pismo podwyższone (tylko dla niektórych drukarek),
u — pismo obniżone,
U — pismo podkreślone,
0-9— sekwencje zdefiniowane przez użytkownika.

I to by było tyle na temat samego edytora tekstu TEXTOMAT PLUS. Przypomnę jeszcze, że na polskim pirackim rynku znajdują się wersje spolszczone programu dla Commodore 64. Niestety mają one „uszkodzone" niektóre funkcje (np. zapisywanie ASCII na dyskietce).

Już wkrótce opiszę program do tworzenia czcionek i parametrów TEXTOMAT ANPASSUNG.

PIOTR LISZEWSKI

Textomat plus 64/128 test
Textomat plus 64/128 test
Textomat plus 64/128 test
Textomat plus 64/128 test
Textomat plus 64/128 test
Textomat plus 64/128 test



Textomat plus 64/128 test
Textomat plus 64/128 test
Textomat plus 64/128 test
Textomat plus 64/128 test
Textomat plus 64/128 test

Przykładowy wydruk rożnych krojów pisma z TEXTOMATA.

TRAGEDIA GRACZA
CZYLI CO SIĘ PSUJE W JOYSTICKU

Do napisania tego artykułu skłoniła mnie lwia część korespondencji od Was, w której z rozpaczą informujecie naszą redakcję o swoich kłopotach z joystickami i związanym z tym wadliwym działaniem komputera. Większość z Was od razu wysnuwa mroczne wnioski jakoby C-64 lub inny komputer miał się uszkodzić. Tymczasem często najciemniej bywa pod latarnią.

Zacznijmy od rzeczy najprostszej. Z chwilą gdy podłączamy joystick, naciśnięcie klawisza F7 wyświetla komunikat np. PRESS PLAY & RECORD, a i znaki wyświetlane po naciśnięciu innych klawiszy nie są zgodne z ich nadrukami. W tym przypadku należy sprawdzić, czy w którymś z joysticków podłączonych do C-64 nie jest włączony autofire. Jeżeli tak, to należy go wyłączyć, a komputer powinien działać poprawnie.

Czasami jednak się zdarza, że jest to coś więcej niż zwykłe włączenie autofire. Trzeba wtedy zagłębić się w konstrukcję joysticka i dowiedzieć się co w nim „piszczy". Ostatnio przeżyliśmy prawdziwą powódź nowych joysticków o wymyślnych kształtach, prześcigających się jakością wykonania oraz długością „życia" podaną w postaci cyfr na pudełku. Jednakże, jak ktoś kiedyś trafnie stwierdził, nie ma rzeczy trwałych, więc nawet najlepszy i najdroższy joystick ma prawo czasem się popsuć.

Warto nadmienić, że joysticki dzielimy na dwie podstawowe grupy:

a) joysticki wyposażone w tzw. mikrostyki, nazywane mikroprzełącznikami lub „switchami",
b) pracujące w oparciu o krzyżak ze sprężynującej blaszki, lub blaszki w kształcie kopułki.

Należałoby utworzyć jeszcze trzecią grupę, do której zaliczyłbym polskie joysticki MAT, których działanie oparte jest na gumkach przewodzących.

Pierwsza grupa jest obecnie najchętniej kupowana, jako że są to joysticki czułe i gra się nimi wygodnie. W joystickach tego typu spotyka się dwa główne rodzaje mikroprzełączników: tzw. mikrostyki kryte (RYS. 1) oraz nowy, trwały i prostej konstrukcji (wykonany specjalnie dla potrzeb branży joystickowej) mikroprzełącznik rozbieralny (RYS. 2).

Pierwszy rodzaj mikroprzełączników spotyka się obecnie w niewielu joystickach (np. Commander, Top Star). Ich główną wadą jest „nienaprawialność" (uszkodzony mikroprzełącznik trzeba zamienić na sprawny). Drugi rodzaj mikroprzełączników jest niebywale prosty w konstrukcji dzięki czemu można łatwo i szybko wymienić uszkodzoną część (najczęściej jest to sprężynka, stanowiąca najsłabszy punkt joysticka).

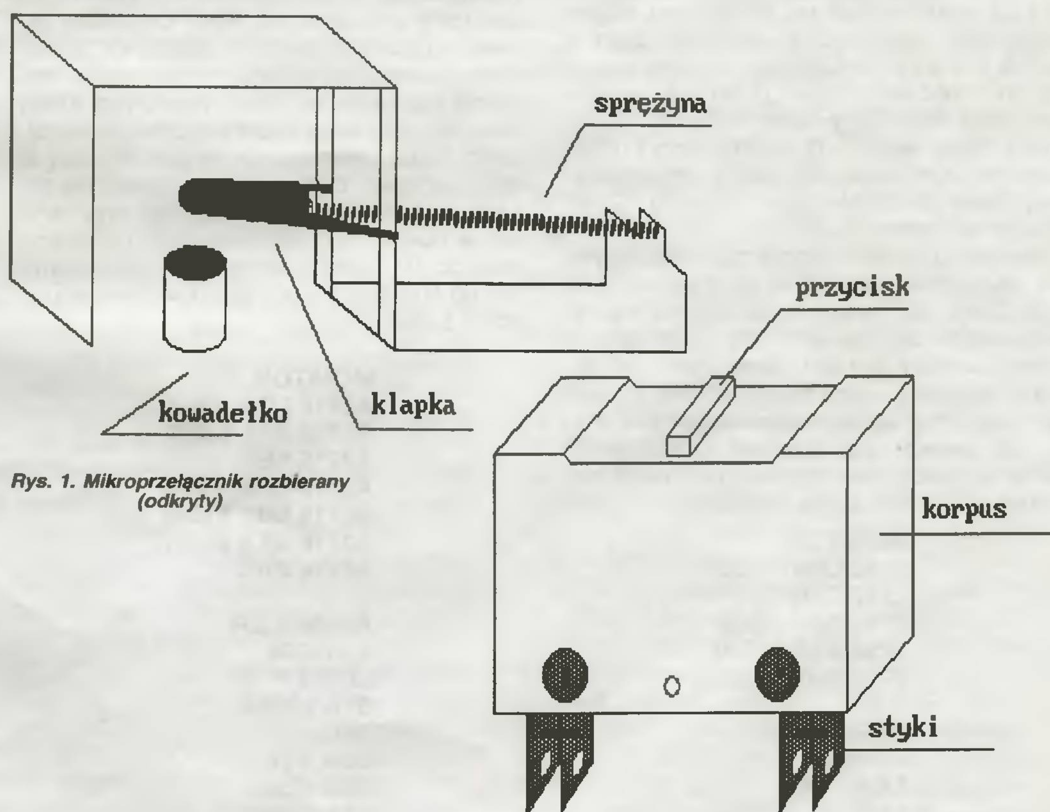
Druga grupa joysticków jest, można śmiało powiedzieć, na wymarciu. Zaliczają się do niej najtańsze josticki (np. QuickShot II). Główną wadą tych joysticków są blaszki stykowe, które w miarę używania pękają lub odkształcają się uniemożliwiając prawidłowe działanie drążka.

Na giełdzie oraz w niektórych sklepach komputerowych możliwy jest zakup sprężynek oraz blaszek stykowych. Niektóre sklepy komputerowe zawierają w swojej ofercie przeróbkę joysticków z blaszkami na mikroprzełączniki co jest najlepszym wyjściem w przypadku uszkodzenia takiej blaszki.

Wreszcie ostatnie z najczęstszych uszkodzeń: pęknięcie przewodu wewnątrz kabla joysticka. Jest to uszkodzenie niekiedy trudne do zlokalizowania jako, że kable mają długość powyżej 100 cm. W joysticku może nie działać wtedy jeden z kierunków, przycisk FIRE, czasem można zaobserwować brak reakcji na jakiekolwiek działanie (przerwanie przewodu „masa"). W takich przypadkach przewód pęka tuż przy wtyczce lub przy wyjściu z obudowy joysticka. Należy wtedy odciąć kabel, ściągnąć izolację z przewodów i przy pomocy np. omomierza sprawdzić przepływ prądu przez żyły. Operację skracania należy powtarzać dotąd, aż uzyskamy przepływ prądu. Należy następnie przylutować odpowiednie przewody (kolory) w odpowiednie miejsca na stykach joysticka. Nie podaję tutaj kolorów i ich oznaczeń, gdyż istnieje wiele standardów kolorowych zależnie od producenta.

Mam nadzieję, że rozwiałem wątpliwości przynajmniej części czytelników, którzy zderzyli się z przykrą rzeczywistością uszkodzenia joysticka.

PIOTR LISZEWSKI



Rys. 1. Mikroprzełącznik rozbierany (odkryty)

Rys. 2. Mikroprzełącznik kryty (obudowany)

Asembler dla każdego (IV)

KOMPUTER — ZNACZY LICZYDŁO!

Jak zdążyłem już nadmienić, słowo *computer* w języku angielskim oznacza mniej więcej tyle co liczydło. Sugerowałoby to istnienie możliwości liczenia na naszej maszynie. Jak już zapewne wiesz, liczenie w BASIC problemem nie jest. Wiemy też, że wszystko, co da się zrobić w BASIC, da się też zrobić w asemblerze. I rzeczywiście, dodawać w asemblerze można i temu też poświęcimy dzisiejsze spotkanie.

Nasz komputer jest jednak zbyt cwaną maszyną, by po prostu wziąć dwie liczby, dodać je i wstawić w jakieś miejsce, z którego łatwo je będzie wyciągnąć. Operacja dodawania przebiega w C-64 w następujący sposób: komputer bierze podaną mu liczbę, traktując ją jako argument lub też adres, pod którym szukać ma argumentu. Po ustaleniu, czym w końcu jest argument, dodaje do niego aktualną zawartość akumulatora i UWAGA! — aktualny stan flagi Carry (przeniesienie). Otrzymaną liczbę wstawia do akumulatora i odpowiednio przestawia flagi Carry (przeniesienie), Zero (zero), Negative (ujemne) i oVerflow (przepełnienie).

Instrukcja, o której ciągle mówimy, nazywa się ADC. Tłumaczy się to na angielski ADd with Carry, zaś na polski: dodawanie z przeniesieniem. Specjalną cechą dodawania z przeniesieniem jest fakt, że do wyniku działania dodaje się jeszcze zawartość flagi C. Jeśli więc chcemy, by wynikiem dodawania było to, co zwykle, bez żadnych dodatkowych sensacji, musimy wyzerować znacznik C rozkazem CLC. ADC działa więc tak:

```
MONITOR
2710 LDA #$02
2712 CLC
2713 ADC #$02
2715 STA $2800
2718 BRK

ASEMBLER
*=10000
LDA #$02
CLC
ADC #$02
STA $2800
BRK
```

Jeśli zadasz sobie trud wpisania i uruchomienia programu (G2710, SYS 10000 albo strzałka w lewo, 3 i S), to komputer zada sobie trud dodania dwóch do dwóch i wstawienia wyniku do komórki $2800, czyli (10240). Po zakończeniu pracy programu (chyba niezauważalnie krótkiej) możemy sprawdzić efekty za pomocą np. PRINT PEEK (10240). Jest 4? Powinno...

Czym jednak byłoby dodawanie bez odrobiny odejmowania? Na szczęście, 6502 ma także odejmowanie. Jest za nie odpowiedzialna instrukcja SBC (SuBstract with Carry czyli odejmowanie z przeniesieniem). Działa ona dokładnie odwrotnie niż ADC. Odwrotnie też interpretuje wskaźnik C. To znaczy od zawartości akumulatora odejmuje argument a następnie odejmuje jeszcze 1, jeśli C jest równy zeru, zaś jeśli flaga C jest zapalona, ta dodatkowa liczba odejmowana nie jest. Musimy to więc pamiętać, by znacznik przeniesienia zapalić rozkazem SEC (ciągle więc odwrotnie niż w dodawaniu). Dla przykładu odejmijmy więc od 10 liczbę 7, którą uprzednio wstawimy do komórki $2800. Wynik wstawimy pod adres $2801.

```
MONITOR
A2710 LDA #$07
A2712 STA $2800
A2715 SEC
A2716 LDA #$0a
A2718 SBC $2800
A271b STA $2801
A271e BRK

ASEMBLER
*=10000
LDA #7
STA $2800
SEC
LDA #10
SBC $2800
STA $2801
BRK
```

Sprawdźmy teraz wynik w komórce 10241. Jest 3? Jeeest! Dla małego eksperymentu zmień jeszcze rozkaz SEC na CLC. I co, jest 2? Taaak!

Czy do tej pory nie zaskoczył, zirytował lub choćby zdziwił Cię fakt, że przy dodawaniu i odejmowaniu zawsze używany jest wskaźnik przeniesienia? Wskaźnik ten służy nam do uproszczenia wykonywanych operacji podczas dodawania/odejmowania dwubajtowego. Skomplikowanie dodawania jednobajtowego jest nieznaczne — wstawiamy tylko jeden rozkaz, który byłby zbędny, gdyby istniało dodawanie bez przeniesienia. Natomiast gorzej byłoby, gdyby nie istniało dodawanie z przeniesieniem. Dodawanie dwubajtowe rozszerzyłoby się ponad miarę — trzeba by dodawać skoki warunkowe, odniesienia do flagi C, jeszcze jedno dodawanie, gdyby flaga ta była ustawiona. Oczywiście, zawsze można by powiedzieć, że projektanci mogli wstawić OBIE te instrukcje. Byłoby to jednak sprzeczne z ogólną filozofią procesora (szybki, mała ilość rozkazów). Dzięki temu właśnie, mimo powolniejszego zegara, procesor ten okazał się lepszym od np. Z80, znacznie bardziej skomplikowanego i zagmatwanego a przez to — wolniejszego.

Czyli — musimy się po prostu przyzwyczaić, żeby przed każdym dodawaniem wyzerować (CLC), a odejmowaniem — ustawić (SEC) znacznik przeniesienia. I już.

Przeniesienie jest bardzo przydatne gdy dodajemy do siebie dwie liczby dwubajtowe (ew. przewidujemy, że wynik naszej operacji przekroczy 255). Musimy wówczas postąpić tak:

1. Pierwszy ze składników dzielimy na młodszy i starszy bajt. Młodszy wstawiamy do komórki $2800, starszy — do $2801.
2. Drugi składnik również dzielimy na bajty. Młodszy kierujemy do $2802 a starszy — do $2803.
3. Wygaszamy wskaźnik przeniesienia (CLC).
4. Dodajemy zawartość komórki $2800 do

zawartości komórki $2802 a wynik odsyłamy do $2804.
5. Dodajemy $2801 do tego, co znajdziemy w komórce $2803. Wynik trafia do $2805.

Wtedy wynik składa się z dwóch bajtów — młodszego ($2804) i starszego ($2805). Znaną już nam metodą przerabiamy to na liczbę dwubajtową i odchodzimy w szczęściu. Przedtem jednak zwróć uwagę, że przed drugim dodawaniem wcale nie zmienialiśmy flagi C. Jeżeli wynik pierwszego dodawania będzie mniejszy niż $ff (255), to znacznik C pozostanie zgaszony i druga operacja przebiega normalnie. Jeżeli jednak liczba ta zostanie przekroczona, to „w pamięci" (jak przy pisemnym dodawaniu) zostaje 1, które trzeba dorzucić do dodawania wyższego rzędu. „Pamięcią" tą, którą na papierze symbolizuje zazwyczaj jedynka w kółeczku, jest tutaj właśnie C. Jeżeli pierwsze dodawanie je ustawi, drugie będzie musiało je uwzględnić. Może się też zdarzyć, że także po drugim dodawaniu znacznik C będzie zapalony. Wtedy musimy traktować wynik jako liczbę trzybajtową, której siedemnastym bitem jest flaga przeniesienia.

Oto zaś program na dodawanie liczb dwubajtowych. Polecam go Twojej szczególnej uwadze:

```
MONITOR
A2710 CLC
A2711 LDA $2800
A2714 ADC $2802
A2717 STA $2804
A271a LDA $2801
A271d ADC $2803
A2720 STA $2805
A2721 BRK

ASEMBLER
*=10000
CLC
LDA $2800
ADC $2802
STA $2804
LDA $2801
ADC $2803
STA $2805
BRK
```

Po udanym wpisaniu i przetestowaniu programu proponuję choć chwilę poeksperymentować i zbadać, czy wszystko się zgadza. Jeśli tak — przystąp do napisania analogicznego programu wykonującego operację odejmowania dwubajtowego. Ha! Prowadzimy tu przecież w końcu kurs PROGRAMOWANIA a nie tylko uważnego przepisywania programów. Mam nadzieję, że zadanie to zbytnio nie przeciąży Twojego napiętego harmonogramu dnia i już za miesiąc porozmawiamy o zwijaniu i rolowaniu bitów. (Ha! Będzie więc na co czekać!).

cdn.

Komisarz Policyi Prostey Powiatu Staszowskiego

BARTEK KACHNIARZ

RAPORT

EDYTOR PL

Jak wiadomo Commodore 64 jest jednym z najlepiej oprogramowanych komputerów domowych. Obok potężnej baterii gier znajduje się także sporo programów użytkowych, w tym także edytorów tekstu. Większość „komodorowców" zapytana o to, z jakiego korzysta programu do pisania tekstów, podałaby tytuły pachnące zagranicą (Easyscript, Textomat, Printfox, Fontmaster), nie wiedząc, że powstał program tego typu napisany przez Polaków. Owym programem jest EDYTOR PL autorstwa dwóch znanych ludzi, twórców WARSAW BASIC, K. Gajewskiego i B. Radziszewskiego. Śmiem twierdzić, że program ten czasami wręcz przewyższa inne znane mi edytory tekstu na C-64, a żeby nie być gołosłownym, podaję kilka informacji na temat tego narzędzia.

EDYTOR

Po uruchomieniu programu ukazuje się menu główne, w którym możemy sobie wybrać sposób edycji tekstu. Można więc wczytać gotowy tekst i kontynuować jego edycję, ale można także utworzyć nowy dokument. Następnie przechodzimy na planszę edycyjną. Pierwsze wrażenie nie jest zbyt przyciągające ze względu na szaroburą kolorystykę. Na szczęście autorzy wzięli to pod uwagę i istnieje możliwość zmiany koloru tekstu, komentarzy i ramki. W bardzo prosty sposób można też zmieniać ustawienie marginesów (podobnie, jak to ma miejsce w edytorze CHIWRITER na IBM).

EDYTOR PL dysponuje oczywiście polskimi literami. Tu słowa uznania dla autorów: program ma wszystkie literki „z ogonkami", i małe, i wielkie, co trudno było uświadczyć w innych edytorach. Co więcej, polskie znaki diakrytyczne bardzo łatwo uzyskać po naciśnięciu klawisza CTRL i żądanej litery. Ponadto przy pomocy klawisza CTRL możemy wykonywać obróbki tekstu typu centrowanie, wytłuszczanie druku czy dołączenie obrazu graficznego (podobną funkcję jak CTRL pełni klawisz COMMODORE LOGO KEY).

Troszkę niewygodna jest opcja wstawiania liter. Nie można na stałe włączyć trybu INSERT tak, jak np. w edytorze TEXTOMAT 64. Gdy chcemy wstawić coś pomiędzy wyrazy lub litery, musimy nacisnąć klawisz F7 (pozostała część tekstu zniknie), następnie wpisać to, co ma być wstawione i ponownie nacisnąć F7. Bardzo dobrze natomiast została pomyślana opcja zaznaczania (MARK) i co za tym idzie — operacje na blokach tekstowych (cięcie, przenoszenie, kasowanie itp.).

EDYTOR PL pozwala tworzyć dokumenty o objętości 33427 znaków (427 linii po 75 znaków = ok. 15,5 strony). Ma ponadto dodatkowe strony dla użytkownika. Jest to stopka, nagłówek i strona robocza. Na stronie roboczej można zapisywać notatki i komentarze lub fragmenty tekstu, które mają być dołączone do głównego dzieła. Strona robocza oczywiście nie jest drukowana, pełni funkcję pomocniczą.

Edytor umożliwia wydruk przy pomocy drukarek: EPSON, SEIKOSHA SG-10/15, STAR NL-10/15, LC-10, MPS 803/801 i kompatybilne. Mogą jednak wystąpić pewne kłopoty w wydruku wytłuszczonego pisma w trybie tekstowym (co jest wadą dość często spotykaną w wielu innych edytorach tekstu).

Edytor przystosowany jest do współpracy ze stacją dysków. Nie zapomniano także o posiadaczach magnetofonów — EDYTOR PL współpracuje także i z tym urządzeniem.

Tekst napisany pod EDYTOREM PL może być czytany przez edytor tekstu VIZAWRITE CLASSICS 128 w trybie 80 znaków. Kolejna cenna zaleta: edytor zapisuje na dyskietce format tekstowy CHIWRITERA (PC XT/AT) łącznie z polskimi literkami! (oczywiście po dokonaniu translacji pliku przez program BIG

BLUE READER lub specjalny program do konwersji plików na C-64 także napisany przez duet Radziszewski–Gajewski). I tyle na temat samego edytora.

KOREKTOR I POMOCNIK KOREKTORA

W komplecie z EDYTOREM PL sprzedawana jest dyskietka z tzw. KOREKTOREM PL. Jest to oddzielny program, który można wywołać spod edytora. Pozwala on na wyszukiwanie ewentualnych błędów ortograficznych w tekście oraz ich korekcję. Wyrazy porównawcze zawarte są w słowniku.

Czasami jednak się zdarza, że baza słówek jest niewystarczająca. W takim wypadku należy rozszerzyć słowniczek. Tę operację umożliwia nam POMOCNIK KOREKTORA PL. Można za jego pomocą rozszerzać zasób słówek zakładając nowe słowniki. Wadą KOREKTORA PL jest dość wolne działanie, co przy dużej ilości tekstu bywa uciążliwe.

INSTRUKCJA

Jest niestety niezbyt estetyczna (odbitka kserograficzna) i napisana zbyt trudnym językiem dla początkujących (specjalistyczne sformułowania ściśle związane z poligrafią).

TAK PO ZA TYM

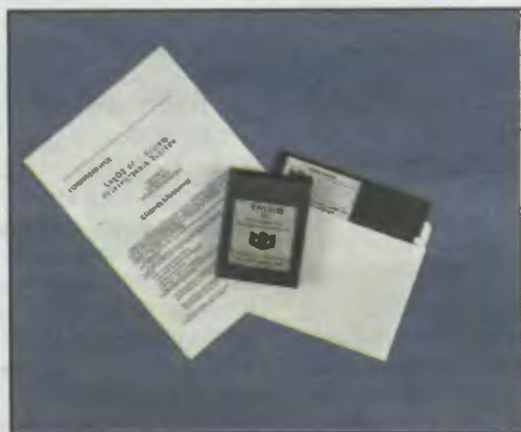
EDYTOR PL sprzedawany jest w postaci modułu podłączanego do C-64/128 przez gniazdo EXPANSION PORT. Niestety takie rozwiązanie ma zasadniczą wadę. Ponieważ port rozszerzenia w komputerze jest tylko jeden, więc po zajęciu go przez moduł z programem EDYTOR PL nie będzie możliwe podłączenie innych kart, np. modułu FINAL III.

I TO BY BYŁO NA TYLE

Podsumowując moje wcześniejsze wywody oceniam program EDYTOR PL jako bardzo dobry i mam nadzieję, że stanie się on często używanym narzędziem dla wielu posiadaczy C-64. Natomiast autorom programu gratuluję wspaniałej pracy i z niecierpliwością oczekuję na kolejne ciekawe programy użytkowe.

PIOTR LISZEWSKI

EDYTOR PL, moduł dla C-64 + dyskietka
DYSTRYBUTOR I PRODUCENT: FUNDACJA EDUKACJI TECHNOLOGICZNEJ, ul. Burdzińskiego 5, 03-480 Warszawa, tel. 180-176.



WADY:
- niewygodna opcja wstawiania,
- nieestetyczna i trudna instrukcja obsługi,
- wolne działanie korektora,
- brak możliwości podłączenia innego modułu w trakcie pracy.

ZALETY:
+ wszystkie polskie litery (małe i wielkie),
+ łatwe zaznaczanie bloków tekstu,
+ proste ustawianie marginesów,
+ możliwość współpracy z magnetofonem,
+ eksport tekstu z CHIWRITERA z polskimi literami,
+ korektor tekstu,
+ prosta obsługa i polskie komunikaty.

OBLICZANIE PIERWIASTKÓW

Za pomocą tego programu można obliczyć pierwiastek dowolnego stopnia z dowolnej liczby (oczywiście prócz liczb ujemnych w przypadku pierwiastków „parzystych"). Program jest bardzo prosty w obsłudze — wszystkie potrzebne komunikaty wyświetlane są na ekranie. Obliczanie pierwiastków następuje poprzez potęgowanie.

```
10 print chr$(147):print:print
20 print "  y  /‾‾
30 print "  ‾\/  x
40 print:print " podaj:
50 input " liczbe pierwiastkowana x ";x
60 input " stopien pierwiastka y    ";y
70 z=x^(1/y)
80 print:print " wynik:";z
90 print:print:print " koniec obliczen (t/n) ?
100 get a$:if a$<>"t" and a$<>"n" then 100
110 if a$="t" then end
120 run
```

RESET

W numerze 5 C&A znajdował się program RESET dla C-64. Nie była to jednak najlepsza wersja tego rozwiązania, ponieważ większość programów w języku maszynowym korzysta z przerwań IRQ (np. ZEGAR TOD, GRABBER 64) podobnie jak program RESET. Mój program eliminuje tę wadę, gdyż korzysta z przerwań NMI, które generuje przycisk RESTORE. Tak więc, aby komputer się zresetował, należy przycisnąć RESTORE. Po zresetowaniu program można ponownie uaktywnić przez SYS 2024.

Jest jeszcze jedna zaleta: mój program zajmuje tylko 14 bajtów.

```
10 for a=2024 to a+13
20 read b
30 poke a,b
40 next
50 sys 2024
60 data 162,243,160,007,142,024,003
70 data 140,025,003,096,076,226,252
```

ARKADIUSZ WIATR

FUNCTION KEYS

Program ten służy do definiowania funkcji dla klawiszy funkcyjnych. Każdemu klawiszowi można przypisać maksymalnie piętnaście znaków, oprócz dwukropka i przecinka. Program składa się z dwóch części:
— programu w języku BASIC,
— procedur w kodzie maszynowym.

Pierwsza część wczytuje dane dla klawiszy funkcyjnych, natomiast druga zmienia wektor przerwań oraz wyświetla podane funkcje.

Aby po naciśnięciu klawisza funkcyjnego funkcja automatycznie się uruchomiła (tzn. aby klawisz RETURN „sam się nacisnął"), należy wpisać:

POKE 494008 + 16*(nr klawisza funkcyjnego — 1) + ilość znaków + 1,13

Program jest bardzo pomocny przy pisaniu długich programów w BASIC — przypisując klawiszom funkcyjnym najczęściej używane instrukcje, np. GOSUB, PRINT, POKE, PEEK i inne, zaoszczędzimy masę czasu.

ARKADIUSZ WIATR

```
100 for a=49152 to 49304
110 read b:poke a,b
120 next
130 for a=49408 to 49538:poke a,0:next
140 for a=1 to 8
150 print"f";a;"= ";
160 input b$
170 if len(b$)>15 then print"max.15 znakow":
    goto 150
180 for c=0 to len(b$)-1
190 d$=mid$(b$,c+1,1)
200 poke 49408+(a-1)*16+c,asc(d$)
210 next
220 next
230 poke 53280,1:poke 53281,1
240 print chr$(147)" function keys"
250 print" krakow 1992"
260 print
270 sys 49152
280 new
290 data 120,173,020,003,174,021,003,141,052,003
300 data 142,053,003,169,025,162,192,141,020,003
310 data 142,021,003,088,096,032,159,255,166,198
320 data 240,100,202,189,119,002,201,133,208,008
330 data 160,000,140,140,192,076,135,192,201,134
340 data 208,008,160,032,140,140,192,076,135,192
350 data 201,135,208,008,160,064,140,140,192,076
360 data 135,192,201,136,208,008,160,096,140,140
370 data 192,076,135,192,201,137,208,008,160,016
380 data 140,140,192,076,135,192,201,138,208,008
390 data 160,048,140,140,192,076,135,192,201,139
400 data 208,008,160,080,140,140,192,076,135,192
410 data 201,140,208,008,160,112,140,140,192,076
420 data 135,192,076,049,234,160,255,200,232,185
430 data 016,193,157,119,002,208,246,134,198,076
440 data 049,234,000
```

GRAFIKA I TEKST

Program ten dzieli ekran na dwie części: graficzną i tekstową. Wielkość części graficznej można regulować przez:
POKE 51509,x
gdzie x to wielkość ekranu razem z ramką (czyli 50).

Program można wykorzystać przy tworzeniu programów, które u góry wyświetlają wykresy czy inne rysunki, a na dole obliczenia. Ponadto można go wykorzystywać do natychmiastowej kontroli własnych programów graficznych w trakcie ich pisania (nie trzeba ciągle zmieniać trybów graficznych, co zajmuje masę czasu). Program w pełni współpracuje z programem FASTPLOT (C&A nr 2/92).

ARKADIUSZ WIATR

```
100 for a=51456 to a+109
110 read b
120 poke a,b
130 next
140 sys 51456
150 data 120,162,033,160,201,142,020,
    003,140,021
160 data 003,169,127,141,013,220,169,
    001,141,026
170 data 208,169,027,141,017,208,169,
    050,141,018
180 data 208,088,096,238,025,208,173,
    017,208,009
190 data 032,141,017,208,173,024,208,
    009,008,141
200 data 024,208,169,226,141,018,208,
    162,070,160
210 data 201,142,020,003,140,021,003,
    076,129,234
220 data 238,025,208,173,017,208,073,
    032,141,017
230 data 208,173,024,208,073,008,141,
    024,208,169
240 data 050,141,018,208,162,033,160,
    201,142,020
250 data 003,140,021,003,076,049,234,
    000,000,000
```

NIEBO II

Listing, który widzisz obok, jest ulepszoną wersją imitacji rozgwieżdżonego nieba, jaka ukazała się w czwartym numerze „C&A" na 25 stronie. Dzięki tej nieco dłuższej procedurze graficznej możesz nie tylko obserwować zapalanie się punktów na ekranie, ale także i gaszenie. Proces ten odbywa się wolniej, niż w poprzednim przypadku, gdy na ekranie ukazywały się jedynie kolejne „gwiazdki".

Linie od 25 do 35 spowalniają zapalanie się punktów, wykonując w tym czasie polecenie gaszenia. Jeżeli chcesz, aby działo się to jeszcze wolniej, musisz wprowadzić pomiędzy wiersze 25–35 linie o treści: GOSUB 105. Ponadto w każdej chwili możesz nacisnąć spację w celu zakończenia i usunięcia programu z pamięci.

Po wczytaniu i uruchomieniu programu FASTPLOT (C&A nr 2), wykonaj NEW, a następnie wczytaj powyższy listing.

ROBERT KULIŚ

```
5 poke 53280,0:poke 53281,0
10 sys 51106
15 sys 51089
20 gosub 75
25 gosub 105
30 gosub 105
35 gosub 105
40 get a$:if a$=chr$(32) then 45
45 goto 20
50 sys 51106
55 sys 51089
60 print
65 new
70 end
75 sys 51127,0,1
80 sys 50990,1
85 x=int(rnd(1)*319)+1
90 y=int(rnd(1)*199)+1
95 sys 51163,x,y
100 return
105 sys 51127,0,1
110 sys 50990,2
115 x=int(rnd(1)*319)+1
120 y=int(rnd(1)*199)+1
125 sys 51163,x,y
130 return
```

PAMIĘTNIKA KANONIERA

URYWEK SIÓDMY

Jeżeli już udało Ci się otrząsnąć z szoku spowodowanego przeze mnie w zeszłym miesiącu (te duszki, które nie wiadomo czemu pojawiały się na ramce) to znak, że jesteś w stanie przyjąć kolejny cios z listy „Tego konstruktorzy nie przewidzieli". Tym razem ujawnię Ci jeden z sekretów autorów demek. Niechętnie dzielą się nim inni, ale prowadzenie stałej rubryki zobowiązuje.

Historia tego efektu jest zbyt dawna i ginie w pomroce dziejów. Niestety, nie są mi znani twórcy za efekt ten odpowiedzialni (jeśli jednak Wam jakoś uda się do takiej informacji dotrzeć, to koniecznie o tym napiszcie!). Ale przystąpmy do rzeczy. Efekt, który mam dziś zamiar zaprezentować, nosi nazwę Flexible Line Distance (Zmienna Odległość Linii — mój wolny przekład), w skrócie FLD. Efekt ten należy do stosunkowo prostych, więc dość często można go spotkać w demach różnej maści i formatu.

Zmienna odległość linii oznacza, że możemy spokojnie zmieniać odległość między dwiema liniami na ekranie. Wygląda to tak, jakbyśmy po prostu odsuwali za obwódkę całą część, która znalazła się pod linią, od której efekt ten zaczynamy. Jak wiemy, do przesuwania ekranu w pionie służy komórka $d011, 53265. Wiadomo też jednak, że za pomocą tej komórki ekran można przesunąć jedynie o siedem linii. Istnieje jednak prosty chwyt, który pozwala nam zwiększyć tę liczbę do dowolnie przez nas wybranej.

Po prostu kiedyś komuś udało się spostrzec, że jeśli do komórki $d011 będziemy wstawiać coraz większe liczby co linijkę, to ekran będzie się coraz bardziej oddalał. Czyli, że w każdej następnej linijce możemy zamiast tego, co na ekranie być powinno, wyświetlać puste miejsce, a właściwą zawartość ekranu odsuwać coraz dalej. Najprościej będzie to robić przez pobieranie kolejnych wartości z rejestru rastra, $d012, „obcięcie" pięciu starszych bitów, włączenie bitów trzeciego i czwartego, a następnie wstawienie tego szatańskiego koktajlu do komórki $d011, czyli przesunięcia pionowego. Dzięki temu w każdej komórce wartość wstawiana do $d011 będzie o 1 większa od poprzedniej, więc nasz zamiar zostanie osiągnięty.

Wadą tego chwytu jest fakt, że w ten sposób nie potrafię rozcinać znaków na części a najmniejszym możliwym odsunięciem są 3 linijki. Żeby zobaczyć, jak wygląda efekt FLD w praktyce, wpisz i uruchom program oznaczony jako LISTING 1. Żeby trochę go uatrakcyjnić, dodałem jeszcze płynne zwiększenie odsunięcia co jedną ramkę (ramką nazywam czas, w którym wiązka elektronów z naszego działa zdąża omieść cały ekran), przez co ekran dość szybko odjeżdża w dół.

Efekt podany w taki sposób ma jednak niewielkie znaczenie przemysłowe, o czym łatwo możesz się przekonać. Odjeżdża cały ekran, co może być pozytywne, lecz nie musi. Oczywiście, można temu zaradzić wykonując podobną procedurę nieco poniżej poprzedniej. Tu musimy odpowiednio wyliczyć, jakie zastosować obsunięcie. Chwila namysłu... Już wiesz? Elementarne, drogi Watsonie. OD = OMAX – OG, czyli obsunięcie dolne równe jest różnicy maksymalnego obsunięcia i aktualnego obsunięcia górnego. I tak właśnie działa LISTING 2. Przed jego uruchomieniem nie zapomnij napisać na ekranie czegokolwiek. Nie musi być mądre, byle zajmowało wiele miejsca.

A z myślą o tych, którym nie wystarcza proste przesuwanie w dół przygotowałem LISTINGI 3 i 4. Zamiast zwiększania obsunięcia wprowadziłem tabelę, do której wpisujemy czego dusza zapragnie a rozum zażyczy. LISTING 4 jest przy tym pomocniczym programikiem w BASIC, który wyręcza nas przy wpisywaniu danych, a dodatkowo robi to dosyć ładnie. Jeżeli chcielibyście zaprojektować własne tablice obsunięć, przydatne będą elementarne wiadomości z trygonometrii. Przypomnę tylko, że funkcja sinus ma okres $2*\pi$, miejsca zerowe w punktach 0, π i $2*\pi$, maksimum (+1) w $0.5*\pi$ a minimum (–1) w $1.5*\pi$. Trzeba też pamiętać, że wszystkie wartości wstawiane do tablicy powinny się zawierać między 3 a 51. Powinno ich być dokładnie 48. To w mojej wersji programu. Jeżeli wprowadzicie jakieś własne zmiany, sytuacja może wyglądać nieco inaczej. Liczba danych jest zależna od kroku w pętli w linii 10. Szerokość sinusoidy — zależy od liczby, przez którą mnożymy wartość funkcji SIN(A) w linii 20. Proponuję też przygotować własne, ciekawe tablice przy użyciu różnych złożeń funkcji. Efekty są.

Jeżeli już jesteśmy przy efektach FLD, nie można nie wspomnieć o kilku interesujących faktach. Po pierwsze, znowu mogą się pojawić czarne pionowe pasy, jak podczas otwierania dolnej ramki. Aby było

je widać, wystarczy zmienić zawartość komórki $3fff. Tu jeszcze jedno małe zastrzeżenie — to działa tylko wtedy, gdy jesteśmy w najniższym banku pamięci.

Ogólnie trzeba zaś przyjąć zasadę, że za pionowe pasy jest odpowiedzialna ostatnia komórka aktualnego banku widzianego przez VIC. Trzeba też Wam wiedzieć, że owe czarne pasy zachowują się względem duszków tak samo, jak grafika, tzn. można duszki za nie schować po zapaleniu odpowiednich bitów w komórce $d01b.

Ponadto warto być może wiedzieć, że linie ekranu mają teraz taką samą długość. Po przejściu FLD nie ma problemu z co ósmą, krótszą linijką. Więcej nie mam nic do dodania, zamykamy konferencję prasową.

cdn.

BARTEK KACHNIARZ



LISTING 1

```
        *= 10000
        lda #<tekst
        ldy #>tekst
        jsr $able
        sei
        lda #$01
        sta $d01a
        lda #$7f
        sta $dc0d
        lda #<irq
        sta $0314
        lda #>irq
        sta $0315
        lda #$32
        sta $d012
        lda #$1b
        sta $d011
        cli
klaps   jmp klaps
irq
r       ldy #$04
        nop
        nop
nlinia  lda $d012
skok    cmp $d012
        beq skok
        and #$0f
        ora #$18
        sta $d011
        dey
        bne nlinia
        inc r+1
        lda r+1
        cmp #$c0
        bne nast
        lda #$03
        sta r+1
nast    lda #$ff
czekaj  cmp $d012
        bne czekaj
        lda #$1b
        sta $d011
        inc $d019
        jmp $ea81
;-----------------------
tekst   .text "hopla"
        .byte 13
        .text "hopla"
        .byte 0
```

LISTING 2

```
        *= 10000
        sei
        lda #$01
        sta $d01a
        lda #$7f
        sta $dc0d
        lda #<irq
        sta $0314
        lda #>irq
        sta $0315
        lda #$34
        sta $d012
        lda #$1b
        sta $d011
        cli
klaps   jmp klaps
irq
r       ldy #3
nlinia  lda $d012
skok    cmp $d012
        beq skok
        and #$07
        ora #$18
        sta $d011
        dey
        bne nlinia
        lda #$35
        sec
        sbc r+1
        tay
        clc
        lda r+1
        adc #$70
poszlo  cmp $d012
        bne poszlo
olinia  lda $d012
jump    cmp $d012
        beq jump
        and #$07
        ora #$18
        sta $d011
        dey
        bne olinia
        ldx r+1
        inx
        cpx #$30
        bne nie28
        ldx #$03
nie28   stx r+1
wyskok  lda #$1b
        sta $d011
        inc $d019
        jmp $ea81
```

LISTING 3

```
        *= 10000
        sei
        lda #$01
        sta $d01a
        lda #$7f
        sta $dc0d
        lda #<irq
        sta $0314
        lda #>irq
        sta $0315
        lda #$34
        sta $d012
        lda #$1b
        sta $d011
        cli
klaps   jmp klaps
irq     ldx tabela
r       ldy tabela,x
        sty $fd
nlinia  lda $d012
skok    cmp $d012
        beq skck
        and #$07
        ora #$18
        sta $d011
        dey
        bne nlinia
        lda #$35
        sec
        sbc $fd
        tay
        clc
        lda $fd
        adc #$70
rty     cmp $d012
        bne rty
olinia  lda $d012
jump    cmp $d012
        beq jump
        and #$07
        ora #$18
        sta $d011
        dey
        bne olinia
        ldx tabela
        inx
        cpx #$31
        bne nie28
        ldx #$01
nie28   stx tabela
wyskok  lda #$1b
        sta $d011
        inc $d019
        jmp $ea81
;----------------------------------------
        *= $2800
tabela  .byte 1
        .byte $03,$04,$05
        .byte $06,$07,$08,$09,$0a,$0b
        .byte $0c,$0d,$0e,$0f,$10,$11
        .byte $12,$13,$14,$15,$16,$17
        .byte $18,$19,$1a,$1b,$1c,$1d
        .byte $1e,$1f,$20,$21,$22,$23
        .byte $24,$25,$26,$27,$28,$29
        .byte $2a,$2b,$2c,$2d,$2e,$2f
        .byte $30,$31,50
```

LISTING 4

```
10 for a=0 to 48 step 3
20 poke $2801+b,(1-sin(a))*48+3
30 b=b+1:next
```



S.C.
Alderan

Alderan S.C.
Korotyńskiego 19a/55, 02-123 Warszawa
telefon: 659-18-21

fax: 628-04-10

S.C. Alderan prezentuje Państwu:

English Teacher

- nasz najnowszy program, przeznaczony dla osób, które pragną dobrze i wszechstronnie opanować język angielski. Program zbudowany jest z kilku modułów:

nauki słownictwa, tłumaczenia zdań, ich dokańczania oraz układania pytań

Każdy z modułów umożliwia wybór jednego z dziesięciu stopni trudności, możliwy jest też wybór tematyki, jak na przykład komputer, zakupy, rodzina, itd. Oprócz tego wszystkiego **English Teacher** oferuje możliwość rozwiązywania angielskich krzyżówek, których jest kilkanaście! Dodatkowo, po każdej poprawnie rozwiązanej krzyżówce program wyświetla hasło, gdy zaś uczący się zbierze je wszystkie - może wysłać je na nasz adres i wziąć udział w losowaniu nagród.

W trosce o uatrakcyjnienie nauki zdecydowaliśmy się także dołączyć do **English Teacher'a** trzy w pełni wartościowe gry: Pacman, Tetris i Puzzle, będące nagrodami za poprawne rozwiązanie testów.

cena promocyjna: 108.000 zł

wymagania sprzętowe: minimum

Prowadzimy detaliczną sprzedaż wysyłkową (odbiorca płaci przy odbiorze, do ceny doliczamy koszty ponoszone na rzecz Poczty). Każdy, kto zamówi tą drogą więcej niż trzy tytuły, otrzyma gratis *Anty-Virus*, a powyżej czterech - program *Notes*.

Poszukujemy poważnych firm chcących zostać naszymi dealerami. Oferujemy korzystne warunki.

Zainteresowani jesteśmy również w nawiązaniu kontaktów ze zdolnymi programistami, a także grafikami, szczególnie specjalizującymi się w animacji trójwymiarowej.

Poza English Teacher'em oferujemy także inne, najwyższej jakości programy użytkowe i edukacyjne:

* ***Poliglota (Lernt mit uns)*** - doskonały program do nauki języka niemieckiego, oparty na najlepszych, powszechnie uznanych materiałach. Podobnie jak wszystkie nasze nowe programy jest maksymalnie rozbudowany i napisany także z myślą o profesjonalnym wyglądzie (tzw. 3D look). Poligiota umożliwia naukę słów, wyrażeń i zwrotów w obu "kierunkach", czyli na pytanie o niemieckie słowo, wyrażenie lub zwrot możemy odpowiadać po polsku albo na polskie - po niemiecku. Materiał podzielony jest na zestawy, każdy po kilkadziesiąt pól, zaś samych zestawów jest 216! Oprócz tego użytkownik ma możliwość samodzielnego definiowania nowych zestawów. Poliglota, jak na dobry program do nauki języka przystało, wyposażony jest oczywiście w syntetyzer mowy, co, zważywszy, że poprawnie wymawia on słowa niemieckie, jest rewelacją!
[komputer: Amiga, cena promocyjna: 90.000 zł]

* ***Panda Pakiet Matematyczny*** - składa się z kilku programów: ***Panda*** - rozwiązywanie równań, rozwiązywanie układów równań dwiema technikami, rozkład wielomianów, działania na wielomianach, wszelkie informacje o funkcji kwadratowej (miejsca zerowe, wyróżnik, wierzchołek paraboli, monotoniczność itd.), oraz program ***Funkcje*** - różnokolorowe, nakładanie jeden na drugi wykresy, analiza wykresów funkcji, kreślenie i wyliczanie pochodnej, aproksymacja pierwiastków, całkowanie (!), wbudowane stałe matematyczne; ***Matematyka*** - pola, obwody figur, wykresy statystyczne, układy równań, macierze, silnia, rachunek prawdopodobieństwa, trójkąt Pascala, działania na wielomianach, krzywe Lissajous, całki, różniczki - to tylko najważniejsze z jego funkcji.
[komputer: Amiga, cena promocyjna 80.000 zł]

Nowości

* ***WordTeacher 2.0*** - wersja 2.0 dobrze znanego i wypróbowanego programu do nauki języka angielskiego (pisowni i wymowy). Posiada wbudowane dwa pełnosprawne słowniki: polsko-angielski i angielsko-polski (35 tysięcy słów). WT 2.0 wykorzystuje syntetyzer mowy, co umożliwia maksymalnie wierne odtwarzanie wymowy angielskiej. Dzięki metodom nauki zastosowanym w tym programie możliwe jest opanowanie z jego pomocą nawet 170 słów w ciągu godziny!
[komputer: Amiga, cena 90.000 zł]

* ***A-Word*** - pierwszy słownik angielsko-polski z prawdziwego zdarzenia, przeznaczony dla komputerów Amiga, bijący konkurencję na głowę zarówno liczbą słów, jak i wykonaniem. Superszybki (napisany w 100% w języku maszynowym), w pełni wykorzystujący wielozadaniowość Amigi. W zależności od upodobań i ilości dostępnej pamięci, możemy uruchomić go "na oknie" lub całym ekranie, możemy też zamknąć jego okno lub ekran, zostawiając go w pamięci jako program "drzemiący", który możemy w każdej chwili uaktywnić kombinacją klawiszy, gdy natrafimy podczas pracy na nieznane słowo. Wśród haseł uwzględniono także wszelkie terminy anglojęzyczne związane z Amigą!
[komputer: Amiga, cena 120.000 zł]

* ***Twój Pierwszy Angielski*** - wspaniała nauka języka angielskiego dla dzieci. Na program składa się 11 scen, w których dziecko ma za zadanie rozpoznać m. in. owoce, kolory i liczby. Nauka odbywa się z wykorzystaniem animacji komputerowej i syntezy mowy, zaś na końcu uczeń przystępuje do egzaminu. O jakości tego programu mogą się Państwo przekonać czytając jego recenzję w czwartym numerze C&A.
[komputery: Amiga/Commodore 64 (dysk i kaseta), cena 180.000 zł]

* ***Ortografia (Gra Słów)*** - zestaw czterech gier rozwijających wyobraźnię, spostrzegawczość, a przede wszystkim wiedzę z dziedziny ortografii (wbudowany słownik zawiera 10.000 słów prosto ze *Słownika Ortograficznego*). Program został skonstruowany z uwzględnieniem wszelkich reguł nauczania, nie jest, na przykład, możliwe uzyskanie na ekranie wyrazu błędnie napisanego - w pamięci utrwala się tylko poprawna pisownia.
[komputer: Amiga, cena 70.000 zł]

* ***Geometria Konstrukcyjna*** - wszystko o geometrii, nauka wszelkich twierdzeń, sposobów rozwiązywania zadań, wszystko w oparciu o przykłady, z wykorzystaniem animacji komputerowej.
[komputer: Amiga, cena 80.000 zł]

* ***Emulator 1.3*** - rewelacja. Za ułamek ceny przeróbki hardware'owej mogą Państwo sprawić, że Wasza Amiga 500 Plus będzie w pełni kompatybilna z poprzednimi modelami. Dzięki emulacji systemu operacyjnego w wersji 1.3, znikną wszystkie Wasze kłopoty z uruchamianiem programów. Zgodność programowa Amigi Plus z uruchomionym *Emulatorem1.3* jest stuprocentowa dzięki temu, że program całkowicie wyłącza system operacyjny w wersji 2.0 i zastępuje go pełnosprawnym Kickstart'em 1.3!
[komputer: Amiga 500+/2000D, cena 99.000 zł]

* ***Pierwsze Kroki*** - program zawierający kilkadziesiąt połączonych z tekstami rysunków, wyjaśniających obsługę Amigi, oprogramowania systemowego oraz sposób podłączania urządzeń zewnętrznych. Zawiera też wyjaśnienia kilkudziesięciu terminów związanych z Amigą. Idealny dla początkujących Amigowców, jak również dla firm sprzedających Amigi (zapewniamy również nalepki na pudełka).
[komputer: Amiga, cena 50.000 zł]

* ***Chemia 2.0*** - najnowsza, znacznie rozbudowana w stosunku do poprzedniej, wersja programu zawierającego wiadomości z zakresu chemii nieorganicznej. Znajdą tu Państwo w formie graficznej wszelkie informacje z tablicy Mendelejewa, jak również wiadomości z dziedziny mechaniki kwantowej. Program umożliwia automatyczne wyszukiwanie wszelkich zależności i podobieństw grup pierwiastków.
[komputer: Amiga, cena 80.000 zł]

* ***Biorytmy 2.0*** - program ten powie Państwu wszystko o Waszej kondycji psychicznej, fizycznej oraz intelektualnej.
[komputer: Amiga, cena 45.000 zł]

* ***Notes*** - bardzo wygodna, elastyczna w swej konstrukcji i prosta w obsłudze podręczna baza danych.
[komputer: Amiga, cena 200.000 zł]

* ***Piórko*** - prosty w obsłudze, ale o dużych możliwościach, dedykowany głównie dzieciom program graficzny.
[komputer: Amiga, cena 80.000 zł]

* ***Anty-Virus*** - zestaw najlepszych programów antywirusowych Public Domain, jakie kiedykolwiek napisano. W zestawie znajdują się między innymi BootX, VirusZ oraz VirusChecker. Dzięki użyciu modemu jesteśmy w stanie oferować Państwu najnowsze wersje tych programów nawet już w kilka dni po ich wypuszczeniu przez autorów.
[komputer: Amiga, cena 55.000 zł]

* ***Zestaw biznesowy*** - fakturowanie, kosztorys, księgowość, magazyn, kadry, płace - wszystko w jednym zestawie. Obliczanie wszelkich podatków, wydruk faktur - tak jak na IBM PC, tylko kilkukrotnie taniej. Wyczerpujący opis - na życzenie. Zapewniamy pełny serwis, łącznie ze sprzedażą kompletnych systemów.
[komputer: Amiga, cena 1.990.000 zł]

ZAPRASZAMY

Jak już informowaliśmy, od numeru 6 C&A rozpoczęliśmy rozpowszechnianie programów Public Domain na Amigę. Zajęci przygotowaniem kolejnych dysków niedopatrzyliśmy sprawy zamieszczenia informacji o ich zawartości. Przepraszamy i uzupełniamy zaległości:

DYSKIETKI C&A NA AMIGĘ — INFO

Dysk nr 1 (C&A 6/92)

VCLI4
Nie wiesz, czym zaskoczyc użytkowników innych komputerów? Ten program Ci pomoże. Dzięki niemu możesz nauczyć swoją Amigę rozpoznawania poleceń wydawanych głosem. Będziesz mógł uruchomić dowolne programy nie dotykając klawiatury czy myszki. Do programu dołączone jest demo. Program wymaga użycia samplera, który można wykonać samemu na podstawie artykułu zamieszczonego w C&A nr 4.

Remap Info
Wszyscy ci, którzy zamienili swoją pięćsetkę na pluskę, docenią zalety tego programu. Pozwala on na automatyczne przemalowanie ikonek rysowanych pod 1.3 na 2.04.

Multi Player v1.17
Jest to uniwersalny „grajek" różnego rodzaju modułów, songów i sampli. Rozpoznaje 12 różnych formatów, w tym wszystkich tzw. trackerów, Oktalyzera, Future Composera, MED'a, sample AudioMastera III i jeszcze parę innych mniej znanych. Program ten nie tylko gra, ale również pokazuje wiele ciekawych rzeczy, zaczynając od informacji o granym module a kończąc na różnego rodzaju spektogramach. Pozwala na pełne kontrolowanie głośności, tempa i balansu granego modułu oraz umożliwia stworzenie longplay'a z kilku modułów. Na dysku znajduje się również moduł Art Coffee do przetestowania Multi Playera.

Copper
Dla miłośników grania mamy również małą propozycję. Jest nią arkanoidopodobna gra Copper. Pomysł stary, ale jary. Gra posiada kilka ciekawych opcji jakich nie miał Arkanoid. W to trzeba zagrać, aby się przekonać.

Inne: dysk uzupełniają pliki download z polskimi znakami do drukarek Star LC10/20 oraz przykładowy sampling wykonany nową, ulepszoną wersją samplera.

Dysk nr 2 (C&A 7/92)

BootX v4.50
Pierwsza nasza propozycja do walki z szerzącym się szybko plemieniem wirusów. Rozpoznaje 351 bootblock'ów, w tym 181 wirusów oraz 52 wirusów plikowych, łącznie z wrednym Saddamem. Sprawdza pamięć i całe dyskietki (bootblock i pliki), spakowane zbiory rozpakowuje przed sprawdzeniem, naprawia szkody wyrządzone przez Saddama. To tylko niektóre z jego możliwości. Dużą zaletą jest możliwość nauczenia BootX'a rozpoznawania nowych wirusów i niestandardowych bootblock'ów. Pozwala na włączenie rezydentnego protektora, który nie pozwoli zainstalować się wirusowi bez naszej wiedzy.

Virus Z v2.19b
Jest to innego rodzaju program antywirusowy. Z chwilą uruchomienia instaluje się on na stałe w pamięci i broni nas przed różnego rodzaju paskudstwem. Sprawdza bootblocki (zna ich 440, w tym 187 wirusów), oraz pliki. Podobnie jak BootX rozpakowuje je przed sprawdzeniem, ale rozpoznaje więcej typów kompresji.

Floppy Disk Base v0.9
Każdemu, kto tonie w stosach dyskietek i godzinami szuka dyskietki z pilnie potrzebnym programem, na pewno przyda się ten program. Pozwala on na stworzenie bazy danych o naszych zasobach dyskowych. Automatycznie wczytuje nazwę dysku i zależnie od potrzeby możemy dodatkowo opatrzyć ją swoim komentarzem. Posiada funkcje sortowania i wyszukiwania. Ilość wprowadzonych rekordów nie jest w żaden sposób ograniczona. Program ten napisał nasz redakcyjny kolega Rafał, nie zapominając oczywiście o pełnej instrukcji w języku polskim.

Pipe Line II
Miłośnicy gry PipeMania będą na pewno zadowoleni z naszej propozycji w sektorze rozrywkowym dyskietki. Niezorientowanym wyjaśnię, na czym polega ta gra. Należy z losowo pojawiających się elementów ułożyć rurociąg, którym popłynie ropa naftowa. Jeśli ropa dotrze do końca rurociągu przed upływem czasu, gra się kończy. Niełatwe zadanie utrudniają liczne niespodzianki w rodzaju np. przerwania rurociągu. Zabawa jest „mocna" i potrafi wciągnąć na długie godziny.

Dysk nr 3 (C&A 8/92)

Power Player v2.7
Druga już propozycja „grajka" na naszych dyskach. Być może nie jest on tak rozbudowany graficznie, jak poprzednik, ale ma swoje zalety. Rozpoznaje 14 formatów modułów, również spakowane Power Packerem. Działa zarówno z CLI, jak i Workbencha, w tym multi wybór (Shift+LMB). Pozwala na dowolne komponowanie longplay'ów z różnych modułów. Do wyboru używa bardzo wygodnej regtools.library. W zestaw wchodzi nowy rodzaj bardzo efektywnego kompresora (pakera) PMC. Power Player odtwarza też moduły spakowane PMC'em.

Slicer
Jest to program do rysowania fraktali. Szerokie możliwości definiowania funkcji matematycznych pozwalają osiągnąć zadziwiające efekty, które dodatkowo można wzbogacić zmianą palety kolorów, czy odpowiednim rzutowaniem. Wygenerowany obrazek można oczywiście zapisać na dysk.

Boot Pic v2
Wszystkim tym, którzy mają dość „żebrzącej rączki" pojawiającej się zawsze po resecie Amigi, dedykujemy ten program. Pozwala on na podmianę owej „rączki" na swój własny obrazek stworzony na przykład DPaintem. Dodatkowo w trakcie pokazywania obrazka może odegrać nasz moduł. Sprytne zainstalowanie programu może być niespodzianką dla nieznających tego „numeru" innych Amigowców.

Scud Buster
Tradycyjnie nie zapomnieliśmy o graczach. Tym razem proponujemy grę strategiczną. Celem gry jest zniszczenie niewidocznego przeciwnika za pomocą rakiet ziemia-ziemia, a jednocześnie obronę swojego terytorium rakietami ziemia-powietrze. Mamy do dyspozycji również radary i AWAC'a, które ułatwiają nam wykonanie tego trudnego zadania.
Inne: pozostałe miejsce zajmują moduły w różnych formatach do odegrania Power Playerem.

Dysk nr 4 (C&A 9/92)

Arq v1.66
Kolejny program z serii oszałamiania przeciwnika, czyli użytkownika innego komputera. Podmienia on standardowe requestery (Disk Write Protect, Volume ?? has read/write error, itd.) na swoje własne, gustowne, trójwymiarowe i w dodatku animowane.

Re Org
Chyba każdy zastanawiał się, dlaczego stacja przy wczytywaniu programów zgrzyta i stuka. W dużej mierze jest to spowodowane działalnością systemu operacyjnego, który przy zapisie „rozrzuca" program po całej dyskietce. Możemy to oczywiście zmienić. Właśnie do tego celu służy Re Org. Optymalizuje on dyskietkę w taki sposób, że programy tworzą jedną całość (co również przyśpiesza wczytywanie programów).

Req Change
Z chwilą gdy Nico Francois napisał cudowną reqtools.library, inni programiści zaczęli wykorzystywać ją w swoich programach. Niestety, wiele starszych programów wykorzystuje nadal inne mniej doskonałe biblioteki jak req, czy arp. Czy można to zmienić? Ależ tak, do tego służy właśnie Req Change. Zmusza on programy do korzystania z reqtools'a.

Rocky v1.0
W kąciku łamacza joysticka mamy klon znanej gry Boulder Dash. Myślę, że tej gry nie trzeba reklamować.

Wszystkie programy opatrzone są naszym komentarzem oraz posiadają pełną dokumentację w języku angielskim.

Przy okazji przypominamy, że cena jednej dyskietki wynosi 15000 zł w redakcji i 21000 zł wysyłkowo. Zamawiając dyskietkę listownie należy wraz z zamówieniem przesłać kopię dowodu wpłaty na konto **SPÓŁDZIELNI „BAJTEK", ul. Wspólna 61, 00-687 Warszawa**. Adres banku: **Bank „Agrobank S.A", ul. Grochowska 262, 04-398 Warszawa**, i numer konta: **470005-1834-131**.

Najlepszym sposobem na zamówienie czegokolwiek w naszej redakcji jest skorzystanie z blankietu drukowanego w C&A. Wszystkich zamawiających drogą listowną prosimy o czytelne wypełnianie zamówień oraz wyraźne zaznaczenie, na jaki komputer chcecie otrzymać dyskietkę. Pozwoli to uniknąć wielu niepotrzebnych kłopotów.

Miłej pracy i zabawy życzy

Redakcja C&A

SUPER SPACE INVADERS

Kiedy na rynku gier telewizyjnych lub, jak kto woli, automatów w roku 1978 pojawiła się gra SPACE INVADERS, nawet sam jej twórca Toshiro Nishikado nie wyobrażał sobie, że jego program może się stać światowym hitem. Sukces pana Nishikado był równocześnie sukcesem firmy TAITO, która tę grę wylansowała. Wkrótce japońska młodzież (i nie tylko), będąca pod wrażeniem ówczesnego przeboju kinowego — GWIEZNYCH WOJEN bez przerwy okupowała salony gier niszcząc nieszczęsne statki kosmiczne poruszające się w szyku. Pomysł gry, przy dzisiejszym poziomie komputerowym może się wydawać śmieszny, lecz wtedy SPACE INVADERS stała się problemem społecznym: dzieci okradały rodziców, aby kupić żetony do automatów, dorośli zawalali robotę, by móc sobie postrzelać, jednym słowem INVADERMANIA.

Firma DOMARK postanowiła przypomnieć dzisiejszemu pokoleniu tę grę w troszkę udoskonalonej formie. Cel jednak pozostał ten sam: rządy obcych poruszajcących się w lewo i w prawo, jednocześnie obniżających swój lot. Ty jako myśliwiec kosmiczny musisz zniszczyć obcych zanim wylądują na dole ekranu.

Oprócz starej fabuły w SUPER SPACE INVADERS pojawiło się wiele nowych elementów. Najeźdźcy przegrupowują się i poruszają się w różnych szykach, zestrzelenie latającego talerza daje nową, o wiele wydajniejszą broń. Po odpartych trzech atakach napotykasz ekran premiowy. Występuje on w dwóch wersjach. W pierwszej występujesz jako dzielny i nieustraszony obrońca krów przed porwaniem (cownapingiem) przez najeźdźców, natomiast w drugiej wersji usiłujesz wykończyć trudnozniszczalnego strażnika końca poziomu.

Mile zaskoczyła mnie instrukcja obsługi (naturalnie w polskim języku), w postaci wcale grubej broszurki, w której dokładnie został wytłumaczony sens gry wraz z rysunkami pomocniczymi. SUPER SPACE INVADERS to dobra gra. Dziękuję Ci DOMARKu, wypróbuję ją na pewno.

PIOTR LISZEWSKI

Dystrybutor:
IPC Computer Group, Warszawa
- Komputer: C-64
- Rodzaj gry: zręcznościowa

CASTLE MASTER

Castle Master to gra labiryntowa, w której na terenie zamku ETERNITY poszukujesz Swojego bliźniaka. Jest on uwięziony w jednej z czterech wież zamku. Zadanie nie jest takie proste, bowiem czychają na Ciebie liczne niebezpieczeństwa. Wnętrze zamku ukrywa sekretne przejścia, tunele i jaskinie. Fascynującą wędrówkę po Eternity utrudniają duchy-strażnicy, których moc znacznie przewyższa Twoją. Aby przeżyć i dojść do celu, musisz zdobywać klucze (do zamkniętych drzwi), eliksiry, a przede wszystkim jedzenie, które zwiększy Twoją cenną siłę.

W Castle Master wymagana jest koncentracja i celność rzutów kamieniami. Nie jest to z pewnością gra dla szybko poddających się, liczy się w niej przede wszystkim cierpliwość. Od celności rzutów zależy Twoje życie. Unieszkodliwiając duchy postaraj się o to, by zajęło Ci to jak najmniej czasu. Przebywanie w jednym pomieszczeniu wraz z duchem odbiera Ci wiele energii, której odzwierciedleniem jest ilość i wielkość kręgów na sztandze w lewym dolnym rogu. Po prawej stronie obok sztangi znajduje się wskaźnik energii duchów — utrzymuj go na jak najniższym poziomie.

Castle Master to gra o dobrej grafice i całkiem niezłej muzyce. Swoje wszystkie ruchy (obroty, bieg, spojrzenie w górę i w dół itp.) możesz wykonywać zarówno joystickiem, jak i za pomocą klawiatury.

ROBERT KULIŚ

Dystrybutor:
IPS Computer Group, Warszawa
- Komputer: C-64
- Rodzaj gry: labiryntowa



Castle Master

HUGO III

No i wreszcie doczekaliśmy się trzeciej części tej wspaniałej gry! Główny bohater gry — HUGO — nie zmienił się, zmieniło się natomiast (acz tylko trochę) Twoje zadanie. Pierwszy etap gry polega na przejściu trudnej drogi usłanej różnego rodzaju pułapkami (czarownica wróciła!). Musisz dokładnie obserwować drogę przed sobą i starać się być zręcznym. W przeciwnym razie możesz szybko, ni stąd ni zowąd dostać w łeb jakimś niesympatycznym kawałkiem gałęzi, który tylko czeka, aby Ci przywalić.

Po drodze czeka Cię jeszcze taka mała niespodzianka, jak wejście na coś w stylu studzienki, która wyrzuca w górę tak że lecisz, lecisz i lecisz... A gdy już dolecisz, będziesz po drugiej stronie ekranu. Wtedy HUGO coś tam sobie pogmera i zacznie tym razem spadać, spadać i spadać... aż stracisz jedno życie (jakżeby inaczej!). Pamiętaj o tym, że grę rozpoczynasz mając do dyspozycji trzy życia.

Drugi poziom nie należy do najtrudniejszych. Polega na zręczności i umiejętności posługiwania się „joyem". Jeśli masz te rzeczy „obcykane", możesz bez obawy zacząć latać. Owo latanie odbywa się w dość trudnych warunkach atmosferycznych, pięknym dwupłatowcem. Podczas lotu uważaj na chmury. I tak: chmura biała jest niegroźna i możesz przez nią spokojnie przelecieć. Natomiast inaczej ma się sprawa z chmurą koloru szarego. Z daleka wygląda niegroźnie, ale gdy się do niej zbliżysz, to zobaczysz całkiem niewinne błyskawice. W tym momencie masz dwa wyjścia: albo zmiatasz tam, gdzie pieprz rośnie, albo jestes na tyle odważny, że lecisz dalej jak gdyby nigdy nic, aż tu nagle zostaje z Ciebie trochę popiołu no i Ty sam,



Hugo

GRY


Kasetowe wersje gier na C-64

na szczęście w całości. Wtedy masz chwilkę czasu na popatrzenie, jak obok ciebie przelatuje sobie dość dowcipny bocian, po czym spadasz, spadasz i spadasz...

Głównym zadaniem na tym poziomie jest zbieranie niebieskich paczek zawieszonych w powietrzu. Pamiętaj, jednak, że nie wolno Ci zbierać paczek koloru czerwonego, bo są to paczuszki z dynamitem! Pamiętaj też, że masz określoną ilość paliwa i jak je zużyjesz, to zaczniesz spadać, spadać i spadać...

Aby uniknąć tych dość chyba nieprzyjemnych niespodzianek, proponuję wylądować i przejść do poziomu trzeciego. Wybacz, ale jestem trochę wredny i nie wyjawię Ci jak to zrobić, powiem tylko, że jest to łatwe. Musisz po prostu pomyśleć, co by było, gdyby...

Ogólnie rzecz biorąc HUGO III jest wspaniały, a pod względem humorystycznym zdecydowanie przewyższa poprzednie wersje. Polecam Wam tę grę zwłaszcza, jeśli macie młodsze rodzeństwo i razem lubicie się trochę pobawić i pośmiać. Nie zaczynajcie grać wieczorem, bo zanim się zorientujecie, będzie już rano!

Co do drugiego poziomu: chciałbym ostrzec wszystkich przed „żonglowaniem" dyskietkami, co może łatwo doprowadzić do szału. Niestety, autorzy gry chyba zapomnieli o tym, że AMIGA jest wyposażona w tylko jedną stację dysków...

OAK

Dystrybutor:
IPS Computer Group, Warszawa

- Komputer: Amiga, IBM
- Rodzaj gry: zręcznościowa, przygodowa.

BAD LANDS

Przenosisz się do złego świata (Bad Lands), gdzie 50 lat temu nastąpiła nuklearna katastrofa, gdzie wyścigi samochodowe stały się bezlitosną walką pomiędzy uzbrojonymi pojazdami. Bierzesz udział w wyścigu, w którym niepowodzenie oznacza śmierć, zaś wygrana — klasyfikację do kolejnego etapu.

Czycha na Ciebie wiele niebezpieczeństw utrudniających jazdę. Jeśli wjedziesz na ciemną plamę oleju, Twój samochód zacznie obracać się, przez co stracisz wiele cennego czasu. Uważaj na różnego rodzaju ruchome zapory, gdyż i tu błąd może spowodować niechciane zakończenie gry. W trakcie wyścigu pomiędzy kolejnymi etapami możesz wyposażyć swój pojazd w turbo (superprzyspieszenie) oraz osłony i broń. Każdy strzał do samochodu znajdującego się przez Tobą ułatwi Ci jego wyprzedzenie, lecz pamiętaj o zbieraniu amunicji, byś nie został bez broni.

BAD LANDS to jedna z lepszych gier symulacyjno-zręcznościowych o dobrej grafice. Przeznaczona jest dla jednego lub dwóch graczy. Zagraj w nią, a z pewnością nie pożałujesz!

ROBERT KULIŚ

CHUCK ROCK

Gdy CHUCK oglądał wiadomości w telewizji, jego żona OPHELIA wieszała pranie. Zauważył to jakiś chciwy koleś z olbrzymią maczugą. Postanowił to wykorzystać. Jak wiadomo w tamtych czasach rządziło jedno prawo: prawo dżungli. Nasz chciwy koleś chwycił czym prędzej za swoją maczugę i uderzył biedną OPHELIĘ, po czym złapał ją za nogi i zaciągnął daleko od domu. CHUCK usłyszał to, niestety za późno. Gdy wyskoczył z domu, OPHELII już nie było. Zdenerwowany CHUCK otrzymał tylko wiadomość od siedzących ptaków na sznurku do suszenia, że zły człowiek, który porwał OPHELIĘ poszedł w prawo. Rozwścieczony CHUCK szybko pobiegł za nimi... no, nie, wcześniej jeszcze przypomniał sobie, że nie ma na sobie żadnego ubrania. Wskoczył w krzaki i zmajstrował sobie — wybaczcie za wyrażenie — gacie. Dopiero potem udał się na poszukiwanie swojej ukochanej.

Taki jest początek gry pod tytułem CHUCK ROCK. Jest to typowa gra przygodowa w dawnym stylu. Samą ideą gry jest długa podróż przez różne straszne krainy i odnalezienie OPHELII. Graficznie program zrobiony jest świetnie, animacja doskonała pod każdym względem. Muzyka, no cóż, nieco gorsza. Całą niesamowitą podróż umila nam superkapela grająca dość ostrego rocka.

Gra nie należy do najłatwiejszych i dlatego dam Wam (jak zawsze) kilka dobrych rad. Po pierwsze: zawsze idź w prawą stronę i nie interesuj się, co jest z lewej strony ekranu. Po drugie: jeśli na ziemi widzisz jakiś kamień, to go podnieś. Później może się okazać, iż bez kamienia nie wskoczysz na wyższą wysepkę i nie przejdziesz dalej. Po trzecie: unikaj wszystkiego co się rusza, ponieważ są to stwory raczej wrogo do Ciebie nastawione, jak np. małpa, która rzuca w ciebie kamieniami, czy stwór, którego jest mi dość trudno opisać. Po czwarte: staraj się nie dotykać groźnie wyglądających rzeczy, bo odbierają Ci one energię. Są to np.: olbrzymie krzaki z dużymi kolcami lub też jakieś ścieki. No i po piąte: postaraj się zbierać tam gdzie to tylko możliwe różne rzeczy, takie jak: łeb od świni, kawałek kurczaka, czy małe serduszko.

Jeżeli lubisz przeistaczać się w niewielką rozwścieczoną istotę, to powinieneś naprawdę poszukać tej gry. Jednakże radzę trochę się pomęczyć i kupić ORYGINAŁ, bo w przypadku pirackich kopii mogą być pewne problemy. Jakie? No, np. po pierwszych dwóch etapach komputer zawiesi się! Nasi hackerzy nie są jeszcze najmądrzejsi i nie każdą grę potrafią, jak to się mówi — „ulepszyć".

Serdecznie namawiam na tę grę. Ja i mój brat szukaliśmy OPHELII od południa do pierwszej trzydzieści w nocy, lecz niestety z marnym skutkiem...

OAK

Dystrybutor:
IPS Computer Group, Warszawa.

- Rodzaj gry: dprzygodowa, zręcznościowa.
- Komputer: Amiga

Dystrybutor:
IPS Computer Group, Warszawa.

- Komputer: **C-64** (kaseta).
- Rodzaj gry: zręcznościowa.

THE SPY WHO LOVED ME

Tym razem rewelacyjny, tajny agent 007 wraz z rosyjskim szpiegiem Anją Amasową, muszą rozwikłać ponurą sprawę zniknięcia dwóch łodzi podwodnych — rosyjskiej i brytyjskiej. Na podstawie mikrofilmów można sądzić, że zostały one skradzione przez szalonego naukowca (skąd my to znamy) Karla Stromberga. Zostajesz więc wysłany na Sardynię w celu znalezienia szaleńca i odzyskania łodzi. Oczywiście nie możesz zapomnieć o superhipertransnowoczesnym samochodzie, samotransformowalnym (taka prywatna amfibia), uzbrojonym po wlew paliwowy w najnowocześniejszą broń (od pracy począwszy).

Cała gra to szalona jazda samochodami i łodziami, w pogoni za Strombergiem. Została podzielona na sceny. Scena 1 to jazda samochodem do portu. Pamiętaj o zbieraniu pieniędzy, które mogą się przydać do zakupu nowej broni oraz o tym, aby oszczędzać przechodniów. Następnie przesiadasz się do motorówki. Cel pozostaje ten sam, z tym, że płyniesz, a nie jedziesz, za to przejechać możesz pływaka, a nie przechodnia. W scenie 2 007 wraz z Anją kierują się do hotelu. Niestety, ściany mają uszy, więc Stromberg dowiaduje się o ich planach i rozkazuje ich zabić.

Bond już dotarł do bazy naukowca, wykonał swoją misję i zawiadomił marynarkę wojenną. Niestety, podstępny Stromberg wziął Anję jako zakładniczkę. Amerykański dowódca wydaje rozkaz zniszczenia siedziby Stromberga. Bond jednak chce uratować Anję. Ma na to godzinę. Czy mu się uda? To zależy już od Ciebie...

Gra przeznaczona jest głównie dla wielbicieli gier zręcznościowych, lecz jestem pewny, że wspaniałą grafiką i muzyką zjedna sobie zwolenników także innych typów gier.

Należy zauważyć, że THE SPY WHO LOVED ME jest nowością. Nawet wśród dużego wyboru pirackiego oprogramowania, oferowanego na giełdzie nie udało mi się znaleźć tej gry. Zatem można śmiało powiedzieć, że firmy oferujące oryginalne oprogramowanie stają się realnym konkurentem dla handlarzy kradzionym oprogramowaniem. Tak trzymać!

PIOTR LISZEWSKI

Dystrybutor:
IPS Computer Group, Warszawa

- Komputer: C-64
- Rodzaj gry: zręcznościowa, przygodowa



The spy who loved me

zostań „włamywaczem”!

(odcinek 2)

ŻYCIE

Skoro możemy już restartować gry, to czas najwyższy zająć się ich ulepszaniem. Najczęściej występującym problemem, zwłaszcza w grach zręcznościowych, jest określona liczba żyć. Programiści z reguły przeznaczają jedną komórkę pamięci na przechowywanie aktualnej ilości żyć. Jednocześnie gdzieś w programie znajduje się tzw. licznik błędów, który zmniejsza (czasami zwiększa) zawartość owej komórki po każdym popełnionym błędzie. Mamy więc do wyboru zwiększanie ilości żyć lub neutralizowanie działania licznika błędów.

Zajmijmy się ulubioną przez zwolenników strzelanin grą COMMANDO, do której odszukaliśmy adres startowy $0850. Po uruchomieniu wciskamy FREEZE i przechodzimy do monitora (opisane czynności dotyczą pracy za pomocą FINAL II, który posiadałem i używałem do włamań). Otwieramy dostęp do całego RAM-u przez wpisanie 04 i wciskamy RETURN. Zwracam uwagę, że wpisujemy LITERĘ a nie cyfrę! W COMMANDO limit żyć wynosi 5 i dlatego używamy opcji H (hunt) monitora, szukamy w pamięci występującej sekwencji bajtów H 0000 FFFF A9 05 i wciskamy RETURN. Sekwencja bajtów A9 05 odpowiada rozkazowi LDA #$05.

Mamy więc w rejestrze akumulatora liczbę równą limitowi błędów. Teraz rozkazem STA $ADRES przeniesiemy ją do komórki pamięci, którą przeznaczyliśmy do przechowywania liczby żyć. Monitor odszuka następujące adresy: $ 08b0, $1ee7, $20cd, $2642, $2c2b, $2db7, $30d7 i 3ec8. Dokonując dis-asemblacji programu (opcja D monitora) od w/w adresów bez trudu ustalamy, że bajt o wartości 5 przechowywany jest w adresach pamięci: $04ff, $0500, $0452,x, $0492,y i $0492,x.

Z reguły ustalenie limitu żyć odbywa się na początku głównej pętli programu. Popatrzmy na jego fragment:

```
08B0 A9 05      LDA #$05
08B2 8D FF 04   STA $04FF
08B5 8D 00 05   STA $0500
```

Jak widać, bajt o wartości 5 umieszczany jest jednocześnie w komórkach $04ff i $0500. Sprawdźmy, czy nasze podejrzenia są słuszne.

Używając asemblera wbudowanego w cartridge wpisujemy A 08B0 LDA #$07 i wciskamy RETURN, przechodząc do Basic i wreszcie przez SYS $0850 (lub SYS 2128 jeżeli ktoś woli) restartujemy grę. Sukces!!! Na ekranie pojawił się napis LIVES 7, a więc pozostało jedynie ustalić czy limit „żyć” przechowywany jest w komórce $04FF czy w $0500. Z poszukiwań adresu startowego pamiętamy, że powyżej niego mamy sporo komórek pamięci wypełnionych rozkazami BRK (Break). Jest to idealne miejsce do wpisania naszej procedurki. Posługując się asemblerem wpisujemy:

```
0830 LDA #$07
0832 STA $0500
0835 JMP $08B8
```

i dodatkowo zmieniamy umieszczony pod adresem $08B5 rozkaz STA $0500 na JMP $0830 oraz przywracamy rozkaz LDA #$05 w lokacji $08B0. Efekt końcowy tych manipulacji wygląda następująco:

```
0830 A9 07      LDA #$07   ;procedura umożliwiająca niezależne
0832 8D 00 05   STA $0500  ;od siebie ustalenie zawartości
0835 4C B8 08   JMP $08B8  ;komórek $04FF i $0500
0850 78         SEI        ;adres startowy
.
.
.
08B0 A9 05      LDA #$05
08B2 8D FF 04   STA $04FF
08B5 4C 30 08   JMP $0830
08B8 A9 A5      LDA #$A5 ; dalszy ciąg programu
.
.
.
```

Po uruchomieniu gry ponownie liczba żyć wynosi 7. Wynika z tego, że ilość żyć przechowywana jest w komórce $0500 pamięci. W zasadzie informacja ta wystarcza już do „ulepszenia" gry. Wpisując pod adres $0830 rozkaz LDA #$FE otrzymamy 254 „życia", co powinno wystarczyć do ukończenia gry. Jeżeli i teraz nie osiągniemy sukcesu, to trzeba znaleźć i zneutralizować licznik błędów, który sprawdza zawartość komórki $0500 zmniejszając o 1 jej zawartość po każdym popełnionym błędzie.

Generalnie rzecz biorąc lista rozkazów mikroprocesora 6502/6510 zawiera cztery rozkazy, które umożliwiają zmniejszenie zawartości komórki pamięci o 1. Są to rozkazy DEC, DEX, DEY i SBC. Rozkazy DEC, DEX i DEY robią to bezpośrednio, natomiast użycie rozkazu SBC wymaga współpracy rejestru akumulatora. Poniższy przykład powinien wyjaśnić różnice w działaniu tych rozkazów (pomijamy DEX i DEY) oraz zapoznać Cię z typowymi konstrukcjami licznika błędów.

a)

LDA #$06	;	6 do akumulatora	Limit „żyć" wynosi 6, informacja ta przechowywana jest w komórce $1000
STA $1000	;	zawartość akumulatora przechowana w komórce $1000	
DEC $1000	;	zmniejszenie o 1 zawartości komórki $1000	Po każdym błędzie ich limit jest zmniejszany o 1
LDA $1000	;	zawartość komórki $1000 do akumulatora	Sprawdzenie, czy wyczerpaliśmy limit błędów, jeżeli tak, to program skacze do procedury końca gry, umieszczonej od adresu $4e17
CMP #$00	;	porównanie zawartości komórki $1000 z 0	
BEQ $4e17	;	gdy równa, skok do $4e17	

b)

LDA #$04	;	4 do akumulatora	Limit „żyć" wynosi 4, informacja ta przechowywana jest w komórce $1000
STA $1000	;	zawartość akumulatora przechowana w komórce $1000	
LDA $1000	;	zawartość komórki $1000 do akumulatora	Po każdym błędzie ich limit jest zmniejszany o 1, sprawdzenie czy wyczerpaliśmy limit błędów, jeżeli tak, to program skacze do
SEC	;	znacznik C=1	

SBC #$01	; zmniejszenie o 1 zawartości akumulatora	procedury końca gry, umieszczonej od adresu $5ea3
STA $1000	; wynik przechowany w komórce $1000	
BCC $5ea3	; gdy w wyniku działań znacznik C=0, skok do adresu $5ea3	

c)

LDA #$08	; 8 do akumulatora	Limit „żyć" wynosi 8, informacja ta przechowywana jest w komórce $15
STA $15	; zawartość akumulatora przechowana w komórce $15	
LDA #$01	; 1 do akumulatora przechowana w komórce $2000	Przygotowanie dekrementacji zawartości komórki $15 o 1
LDA $15	; zawartość komórki $15 do akumulatora	Po każdym błędzie ich limit jest zmniejszany o 1, sprawdzenie czy wyczerpaliśmy limit błędów, jeżeli tak, to program skacze do procedury końca gry, umieszczonej od adresu $3d12
SEC	; znacznik C=1	
SBC $2000	; zawartość komórki $2000 odjęta od akumulatora	
STA $15	; wynik przechowany w komórce $15	
BCC $3d12	; gdy w wyniku działań znacznik C=0, skok do adresu $3d12	

Jak nietrudno zauważyć, najprostszym i jednocześnie zajmującym najmniej pamięci jest sposób kontroli limitu błędów z przykładu a). Z własnej praktyki wiem, że jest to sposób najczęściej stosowany przez programistów. Wróćmy więc do Commando i sprawdźmy, jak zmniejszana jest zawartość komórki $0500. Po chwili monitor znajduje nam adres $0996 w którym zaczyna się rozkaz DEC $0500, któremu odpowiada sekwencja bajtów ce 00 05. Spróbujmy zastąpić go przez 3 rozkazy NOP (No OPeration) i uruchomić grę. Eureka! Nieustraszony komandos odradza się po śmierci a licznik błędów cały czas pokazuje 5. Można strzelać do woli.

Zastąpienie rozkazu DEC $0500 przez 3 rozkazy NOP, aczkolwiek skuteczne, nie jest rozwiązaniem eleganckim, a ponadto, jeżeli zmiany będziemy wprowadzać z poziomu Basic, wymaga wpisywania aż trzech POKE'ów. Jeżeli rozkaz DEC $0500 zastąpimy przez LDA $0500, to również osiągniemy efekt „nieśmiertelności", a w BASIC wystarczy tylko jeden POKE. Jest to chyba oczywiste, jeżeli porównamy zapisy tych rozkazów w asemblerze:

0966 CE 00 05 DEC $0500
0966 AD 00 05 LDA $0500

Przeliczenie z heksadecymalnych na dziesiętne adresów i wartości, które trzeba w nie wpisać, pozostawiam zainteresowanym Czytelnikom. W powyższy lub zbliżony do niego sposób można odszukać „nieśmiertelność" w większości gier, gdyż przedstawioną metodę programiści wykorzystują najczęściej do ustalenia w grze limitu żyć.

cdn.

URAN



Tytuł / Liczba kolejnych zeszytów	3	6	12	po ile egz.
Bajtek	X	60000	120000	
C&A	30000	60000	X	
TOP SECRET	27000	54000	X	

Wpłat dokonywać na konto:
Spółdzielnia "BAJTEK"
Bank "Agrobank S.A."
470005-1834-131
ul. Grochowska 262
04-398 Warszawa

tu zanotuj, co zamówiłeś

Warunki prenumeraty:

- Prenumerata zawarta przed upływem ważności kuponu gwarantuje niezmienność cen
- Przesyłka pocztowa nie wymaga dodatkowych opłat
- Jeżeli w ciągu 2 tyg. od pojawienia się numeru w kioskach przesyłka nie nadeszła, prosimy o kontakt
- Za błędy wynikające z niestarannnego wypełnienia formularza redakcja nie ponosi odpowiedzialności
- Prosimy o staranne i wyraźne zakreślenie odpowiednich ilości egzemplarzy

NINIEJSZYM ZAMAWIAM:

m-c	1	2	3	4	5	6	7	8	9	10
"C&A"										
Dyskietka 5,25" dla C-64/128/16/116/+4										
Dyskietka 3,5" dla Amigi										

Kaseta dla C-64 ☐

IMIĘ: . . . .
NAZWISKO: . . . .
ADRES: . . . .
. . . . (Kod): . . . . .

UWAGI:

Do kuponu dołączam kopię dowodu wpłaty.

NAJPIERW PRZECZYTAJ, POTEM ZAMAWIAJ!

1. Kupon ułatwia i przyspiesza realizowanie Twoich zamówień.
2. Kupon bez dowodu wpłaty (lub kopii) jest tylko nic nie wartym papierkiem. Zawsze, zanim wyślesz do nas zamówienie, upewnij się, czy włożyłeś do koperty **kupon i kopię dowodu wpłaty.**
3. Zaznacz krzyżykiem w odpowiednim miejscu to, co zamówiłeś. Jeśli Twoje zamówienie jest nietypowe, masz miejsce w rubryce „UWAGI", aby wszystko dokładnie wyjaśnić.
4. Podawaj zawsze **PEŁNY** adres, nie zapominaj o kodzie pocztowym, numerze ulicy itp.
5. Wypełnij kupon **WYRAŹNIE!**

Ceny:

Numery archiwalne "C&A":	12.000 zł/szt.
Dyskietki 5,25" na C-64/128/16/116/+4:	21.000 zł/szt.
Dyskietki 3,5" na Amigę:	21.000 zł/szt.
Kaseta na C-64:	100.000 zł/szt.

UWAGA!

Kaseta kosztuje sto tysięcy, a NIE dziesięć! Zawarte są na niej wszystkie programy dla C-64 publikowane w pierwszych sześciu numerach C&A.

ANKIETA ANKIETA ANKIETA ANKIETA

Na naszą ankietę zamieszczoną w 4 numerze „C&A" odpowiedziało prawie 3.5 tysiąca osób. Dzięki Wam udało się nam przybliżyć do przeciętnego użytkownika komputerów Commodore, mogliśmy się również zorientować w Waszych oczekiwaniach dotyczących naszego magazynu. Zgodnie z obietnicą rozlosowaliśmy pomiędzy Was szereg nagród i upominków o czym za chwilę.

„NIEŚMIERTELNE" C-64

Dość dużym zaskoczeniem dla nas wszystkich był fakt, że prawie 65% ankietowanych to właściciele C-64 (2188 osób). W skład standardowego zestawu wchodzi prawie zawsze magnetofon (występuje on w ponad 90% zestawów), niekiedy ze stacją dysków.

O ile mi wiadomo, większość C-64 sprzętu trafiła do rąk nowych posiadaczy w postaci podarunków i prezentów od krewnych wymieniających sobie sprzęt na bardziej zaawansowany. W ten sposób „wysłużony" C-64 posłuży jeszcze kolejnemu pokoleniu do zdobycia pierwszych „komputerowych" ostróg.

Wspominane pokolenie garnie się do nauki. Ponad 51% czytelników uważa, że w „C&A" należałoby zająć się nauką programowania, publikowaniem programów (47%), opisów oprogramowania (44%), podawaniem tricków (43%) oraz porad (42%). Zupełnie innego zdania są natomiast właściciele sprzętu 16-bitowego, którym (sądząc z wykresu zamieszczonego na rysunku 4) publikowanie programów do wpisania wydaje się niepotrzebne. Jest to moim zdaniem konsekwencja faktu, iż do komputerów 16-bitowych oprogramowania jest w bród i można w nim przebierać jak w ulęgałkach; stąd też zapotrzebowanie na publikacje tego typu jest niewielkie.

Bez niespodzianek odbyło się przy pytaniach „Czy jesteś zadowolony ze swojego komputera" oraz „Czy chcesz zmienić swój komputer na lepszy". Zadowolenie ze sprzętu 8-bitowego wyraziło jedynie 23% czytelników, 58% zaś odpowiedziało na to pytanie negatywnie. Chęć wymiany na lepszy lub bardziej zaawansowany technologicznie komputer wyraziło ponad 78% (!) właścicieli sprzętu 8-bitowego oraz ponad 51% posiadaczy 16-bitowych komputerów Commodore. Moim zdaniem świadczy to o tym, że istniejący sprzęt powoli staje się niewystarczający w stosunku do potrzeb entuzjastów mikroinformatyki.

Największym wzięciem cieszy się obecnie Amiga. Popularną „pięćsetkę" chciałoby kupić w tej chwili prawie 44% ankietowanych; łącznie Amiga zebrała 84.5% głosów (CDTV — 1.1%, natomiast komputery serii PC — 14.5%). Zaznaczam też, że wielu ankietowanych wyrażało gotowość zatrzymania swojego komputera po dokupieniu do niego dysku twardego, rozszerzenia pamięci czy wreszcie dodatkowej stacji dysków.

Największą grupą korzystającą z Commodore są uczniowie — to właśnie u nich jest 2667 tych komputerów. Dość licznie występują również studenci (82), technicy (54), mechanicy (54), elektronicy (40), nauczyciele (37).

Za miesiąc przedstawię typy programów jakie nadesłaliście do nas w ankiecie.

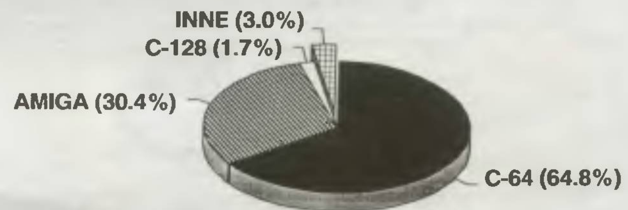
KLAUDIUSZ DYBOWSKI

NAGRODY WYLOSOWALI:

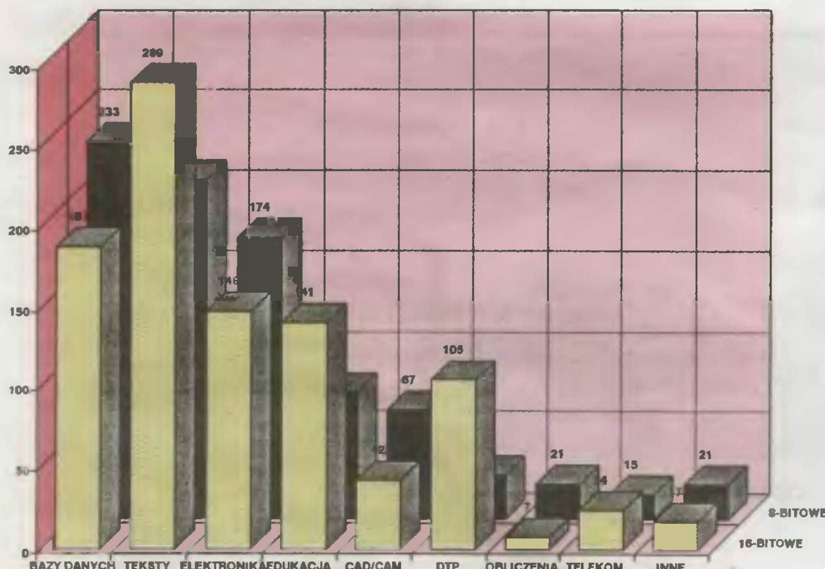
1. **GRZEGORZ BERDZIK,** Stargard Szcz. — moduł WARSAW BASIC
2. **SEBASTIAN PROKOPCZUK,** Wrocław — moduł WARSAW BASIC
3. **TOMASZ KOTLARSKI,** Bielsko-Biała — moduł WARSAW BASIC
4. **MAREK BARANOWSKI,** Reda — moduł WARSAW BASIC
5. **PAWEŁ KIRAGA,** Pionki — moduł HELP PL
6. **WACŁAW SZEWCZYK,** Warszawa — moduł HELP PL
7. **PIOTR ZAJĄC,** Lipsko — moduł MAX PL
8. **KRZYSZTOF ŻAK,** Ogrodzieniec — moduł EDYTOR PL
9. **ARTUR NATANEK,** Oleśnica — moduł WARSAW BASIC + książka „Pierwszy kontakt z programowaniem C-64"
10. **MICHAŁ KASZUBA,** Dąbrowa Górnicza — plecak firmowy Commodore + książka „Pierwszy kontakt z programowaniem C-64"
11. **ALEKSANDER SZUBERT,** Zgierz — plecak firmowy Commodore + książka „Pierwszy kontakt z programowaniem C-64"
12. **BARTŁOMIEJ RYKOWSKI,** Barlinek — książka „Pierwszy kontakt z programowaniem C-64"
13. **ROBERT STUDZIŃSKI,** Nowy Sącz — koszulka firmowa Commodore + książka „Pierwszy kontakt z komputerem C-64"
14. **PAWEŁ ZAWISZA,** Trzebnica — koszulka firmowa Commodore + książka „Pierwszy kontakt z komputerem C-64"
15. **MICHAŁ MUC,** Częstochowa — koszulka firmowa Commodore + książka „Pierwszy kontakt z muzyką i grafiką C-64"
16. **PIOTR PASTERNAK,** Kraków — książka „Moja Amiga" (cz. I i II) + gra POWER MONGER na Amigę
17. **ANDRZEJ KOSMAN,** Radom — gra FATE na Amigę
18. **WOJCIECH BOGACKI,** Tychy — gra STEEL EMPIRE na Amigę
19. **WOJCIECH PLANER,** Bydgoszcz — gra BLACK CRYPT na Amigę
20. **PAWEŁ GAJDOWSKI,** Kołobrzeg — gra ZONE WARRIOR na Amigę
21. **LESZEK KOSMALA,** Złotoryja — gra POWER MONGER na Amigę
22. **MAREK KONOPKA,** Kołobrzeg — książka „Moja Amiga" (cz. I i II)
23. **TADEUSZ BARANIAK,** Mamieczki — moduł EDYTOR PL

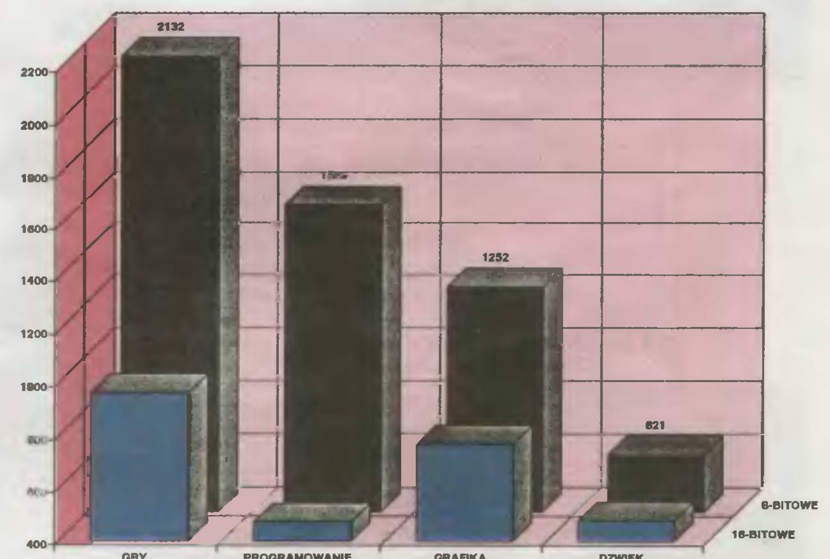
*Fundatorem nagród jest **FUNDACJA EDUKACJI TECHNOLOGICZNEJ** (moduły, koszulki, książki, plecaki) oraz **IPS Computer Group** (gry na Amigę). Dziękujemy!*

COMMODORE U UŻYTKOWNIKÓW (RAZEM 3343 KOMPUTERY)

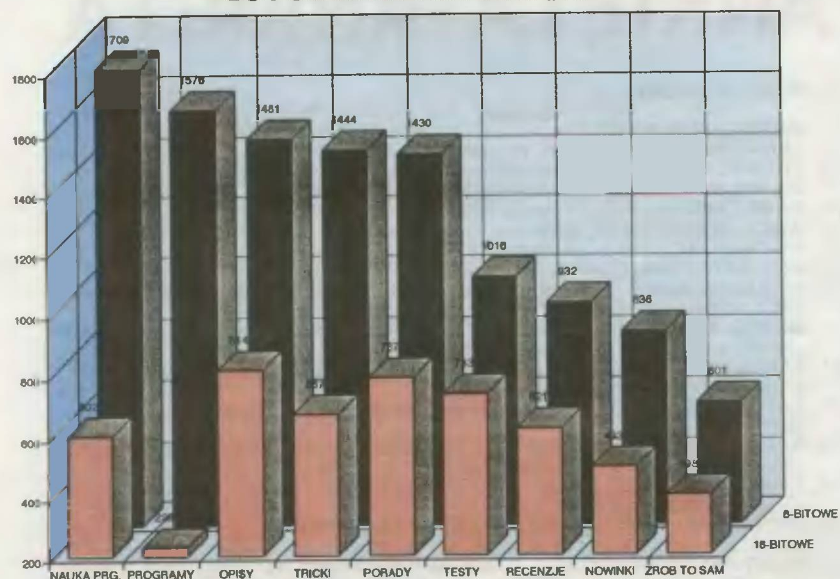


ZAINTERESOWANIA NASZYCH CZYTELNIKÓW 1 2

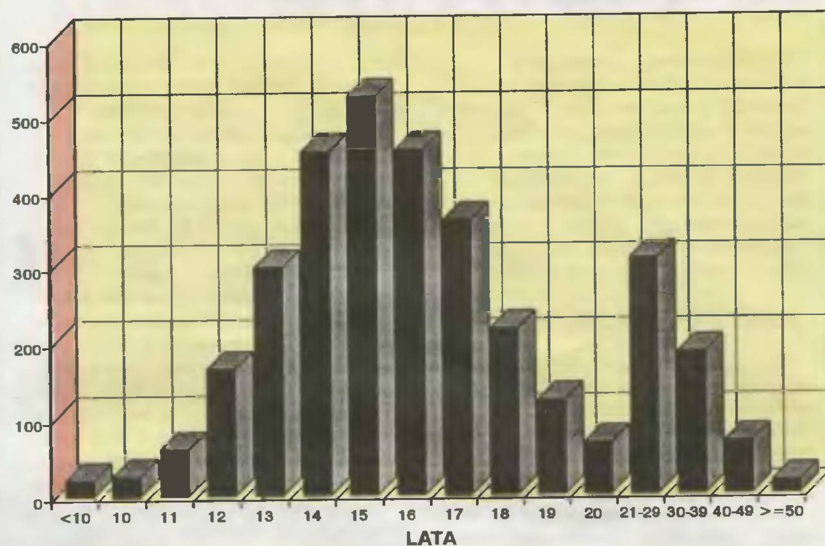




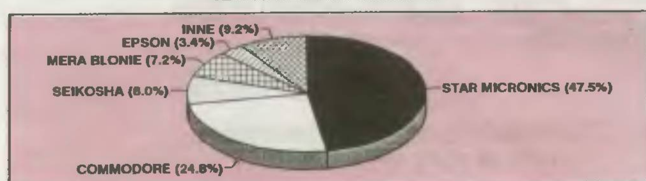
CO POWINNO ZAWIERAĆ „C&A”



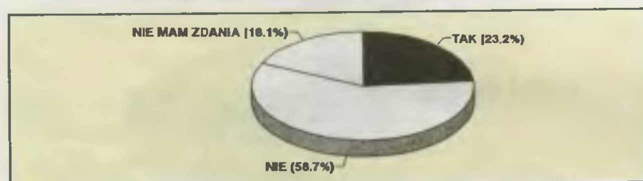
ZAINTERESOWANIE KOMPUTEREM W ZALEŻNOŚCI OD WIEKU



W DRUKARKI WYPOSAŻONYCH JEST 12.4% UŻYTKOWNIKÓW



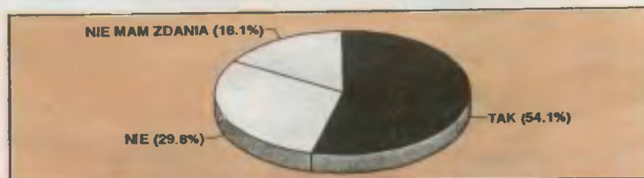
CZY JESTEŚ ZADOWOLONY ZE SWEGO 8-BITOWEGO KOMPUTERA?



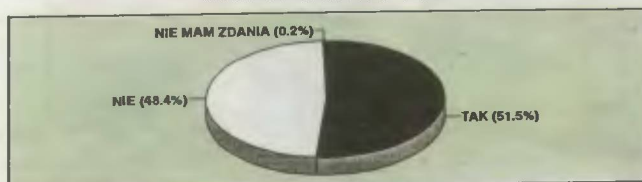
CZY CHCESZ ZMIENIĆ SWÓJ 8-BITOWY KOMPUTER NA LEPSZY?



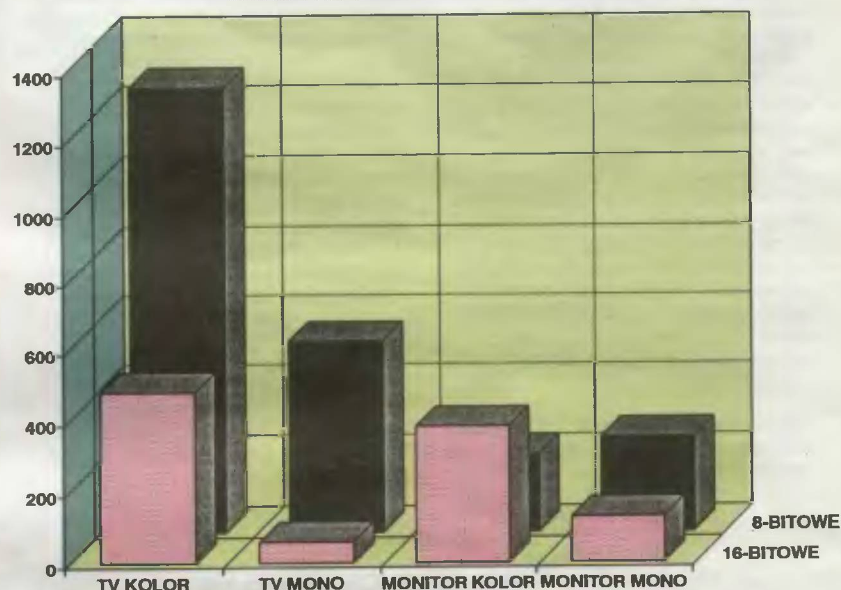
CZY JESTEŚ ZADOWOLONY ZE SWEGO 16-BITOWEGO KOMPUTERA?



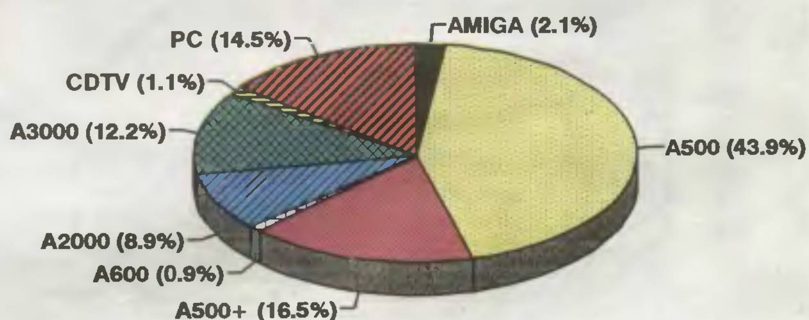
CZY CHCESZ ZMIENIĆ SWÓJ 16-BITOWY KOMPUTER NA LEPSZY?



NA CZYM ODBIERACIE OBRAZ



JAKI KOMPUTER CHCESZ SOBIE KUPIĆ?



JĘZYKI PROGRAMOWANIA

czyli jak dogadać się z maszyną

W poprzednich artykułach opisałem kilka najważniejszych języków programowania. Teraz kiedy wiemy, czym się one charakteryzują, nadszedł czas na zapoznanie się z kompilatorami lub interpreterami tych języków. Zacznijmy od asemblerów. Ponieważ ten temat był już poruszany a łamach naszego pisma, więc dodam tylko po parę słów do wcześniej opisanych asemblerów.

MasterSeka (vl.80) — ma bardzo wygodny edytor i debugger. Wadą natomiast jest to, że program wynikowy ma tylko dwa tzw. hunki, czyli blok kodu i blok danych. Autor: Corsair:SkidRow.

AsmOne (vl.07) — wprost cudowny, pozwala na łączenie programów z tzw. includami, plikami zawierającymi definicje stałych, makrodefinicji i struktur. Wyposażony jest w source debugger. Autor: Rune Gram-Madsen.

Devpac (v3.xx) — w pełni profesjonalny, ale bez 2 MB RAM twardego dysku jego użycie jest problematyczne. Firma: HiSoft.

ArgAsm (vl.0) — najszybszy z asemblerów na Amigę i to jest jego jedyna zaleta. Wymaga 2 MB RAM i HDD. Firma: Argonaut.

AssemPro — prezentuje się nieźle, ale zapisuje pliki źródłowe w niestandardowym formacie. Firma: Oxxi.

A68k (v2.61) — Public Domain asembler napisany przez Charlie Gibbsa w języku C na podstawie kodu źródłowego Briana Andersona w Moduli 2. Jest przeważnie dołączany do kompilatorów C.

ACC — opracowany na podstawie A68k i PD linkera Blinka. Rozpowszechniany przez klub Amiganuts United dla pragnących nauczyć się programować w asemblerze.

Latticce/SAS C — dołączony do tego zestawu asembler sam oblicza offsety funkcji bibliotecznych, jeżeli są one zdefiniowane w programie jako zewnętrzne i dołączymy odpowiednie pliki nagłówkowe.

Manx Aztec C — kolejny asembler dołączony do kompilatora języka C.

ADAPT — bardzo mało znany napisany przez Lake Forest Logic Inc.

Avocet 68000 Family Assembler
Cape 68K 2.52 Assembler
Macro68 V3 (Macro-Assembler)

BASIC

AmigaBasic — interpreter. Dostarczany razem z komputerem (pierwsza zaleta), i książką (druga i ostatnia zaleta). Strasznie wolny, co widać zwłaszcza przy odświeżaniu ekranu i niezbyt „przyjacielski", wręcz „nieprzyjacielski" w niektórych sytuacjach (np. współpraca z buforem). Ale jest strukturalny. Firma: Microsoft.

Cursor (vl.0) — trzyetapowy PD kompilator do programów napisanych w AmigaBasic.

GFA Basic — napisany na podstawie GFA Basic z Atari ST jest z nim zgodny, ale jak przystało na język napisany dla Amigi, tylko jednostronnie (ST→Amiga), bowiem ma rozszerzenia dotyczące menu, tzw. obiektów i obrazków w trybie HAM. Jest to interpreter (szybki), ale na życzenie można otrzymać kompilator jeśli masz zarejestrowaną kopię tego pierwszego. Ma jedną wadę: trzeba samemu zdefiniować wyjście z programu, w przeciwnym razie program będzie się wykonywał do końca świata (Ctrl-A-A) lub awarii zasilania. Firma: Data Media UK.

Hisoft Basic — kompilator. Rozpowszechniany w wersji standard, zgodnej z AmigaBasic, i rozszerzonej, oferującej obsługę gadgetów, sampli itd. poprzez procedury napisane przez programistów Hisoftu. Programy wynikowe są małe, ale do poprawnego działania potrzebują dużej biblioteki hisoftbasic.library.

Amos — najszybszy interpreter jaki widziałem. Początkowo napisany jako szybki język do pisania gier, ostatnia jego wersja pozwala praktycznie na napisanie każdego programu. Wyposażony w wewnętrzny język Amal pozwalający obsługiwać szybkie animacje. Wykorzystanie programów czy procedur asemblera też nie nastręcza problemów. Powstał również kompilator Amosa (napisany także w Amosie). Autor: Francois Lionet.

Blitz Basic — napisany jako narzędzie do pisania gier. Podobno najszybszy Basic.

C

Lattice/SAS C (v5.10a) — najlepszy (moim zdaniem) pakiet do pisania programów w C. Zawiera wszystko co potrzebne: kompilator linker (PD Blink) edytor i source debugger oraz wszystkie pliki nagłówkowe i biblioteki. Napisany według standardu ANSI (American National Standard Institute). Ostatnio powstał Lattice C++ do programowania obiektowego.

Manx Aztec C (v5.0) — następny komercyjny kompilator C dostarczany ze wszystkim, co niezbędne do pisania programów. Również typu ANSI.

Sozobon C — przeniesiona na Amigę wersja najstarszego PD kompilatora C z Atari ST. Napisany według standardu K&R (tj. według książki opublikowanej przez autorów języka — różni się od obecnie przyjętego standardu ANSI sposobem deklaracji i prototypami funkcji).

ZC (vl.01) — zmodyfikowana wersja powyższego kompilatora. W skład pakietu wchodzi asembler A68k, linker i optymizer. Zawiera standardowe pliki nagłówkowe i własne biblioteki. Niestety, jak każdy PD kompilator, nie ma plików i bibliotek niezbędnych do wykorzystania pełnych możliwości Amigi. Trzeba je zamówić u dealera Commodore za około $20.

North C (vl.2) — kolejny PD, K&R kompilator. Mówią, że lepszy od poprzednich.

PDC (v3.33) — jedyny PD kompilator typu ANSI (no prawie).

HCC — najnowsza modyfikacja Sozobona. PD.

GCC i GNU-C — PD C i C++ kompilatory napisane na podstawie kompilatorów dla Unixa. Wymagają HD.

Dice — shareware kompilator. Wersja rozpowszechniana bezpłatnie nie wykonuje działań na liczbach zmiennoprzecinkowych i oczywiście nie zawiera bibliotek i „includów" Commodore.

PASCAL

HighSpeed Pascal — profesjonalny, w pełni zgodny z Turbo Pascalem (I*B) kompilator Pascala. Umożliwia nawet wykorzystanie modułów BGI (dotyczących grafiki). Bardzo szybki. Firma Hisoft.

PCQ — PD kompilator Pascala.
Kick Pascal
Kyan Pascal
MetaComco Pascal

LISP

Metacomco Lisp
OakLisp
Xlisp (v1.6) — PD interpreter napisany przez autora XLispu, pana D Betza. XLisp jest nową, rozwojową wersją tego języka.
AMXLISP.

LOGO

PowerLogo — PD
ALogo — PD. Autor G. Owens

PROLOG

SB Prolog (v2.3.2) — Stoney Brook Prolog
Wisdom Prolog
TinyProlog — PD

FORTRAN

Absoft Fortran 77
Fortran/020
BCF Fortran 77

MODULA 2

TDI modula
M2Amiga
Benchmark Modula-2
M2Sprint
Titan Modula II — Firma Oxxi

FORTH

JForth Pro 3.0
MVP-Forth

INNE

AREXX — jest to język programowania plików skryptowych do komunikacji między programami, w tym uruchamiania jednego programu z drugiego.

Oberon — cokolwiek by to nie było, jest na Amigę.

Draco — hybryda powstała z połączenia C i Pascala.

Smalltalk — PD interpreter języka zorientowanego obiektowo. Podobnie jak C++ i Turbo Pascal v5.5.

Visionary — do pisania gier.
ADL — do pisania gier przygodowych.
Amiga Comal — PD.

Pilot — też nie wiadomo co, ale PD.

Ada — jest to specjalny język do oprogramowywania komputerów nadzorujących różne urządzenia techniczne. Powstała na zamówienie Departamentu Obrony USA, więc wiemy o jakie urządzenia chodzi. Szybsza od asemblerów.

SuperBaseProfesional V4.0 — program ten ma swój własny język programowania, tzw. język aplikacyjny.

I to by było wszystko, co chciałem przekazać na temat języków programowania. Lista ta zawiera wszystkie znane mi (przynajmniej z nazwy) kompilatory lub/i interpretery. Oczywiście, ponieważ wyobraźnia i chęć tworzenia ludzi nie zna granic, lista ta może stać się nieaktualna w ciągu tego tygodnia.

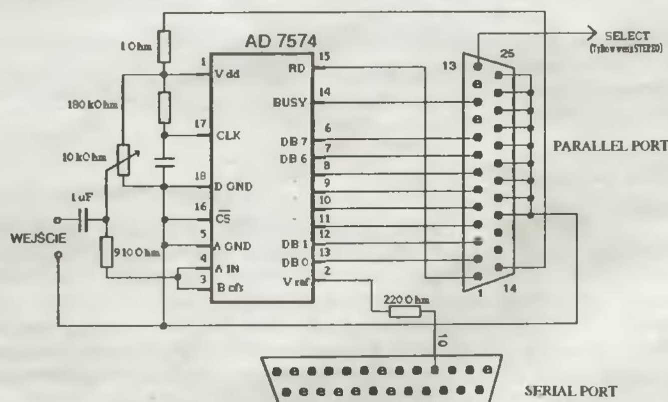
PAWEŁ GALAS
(TEAM ROBOCOST)

PS: PD oznacza Public Domain, czyli program za darmo lub za symboliczną opłatą. Bez ograniczeń co do kopiowania.

ERRATA DO C&A 9/92

W numerze dziewiątym C&A złośliwy chochlik zrobił nam największy kawał w historii, a mianowicie „wyciął" całą stronę z listingiem do tekstu Bartka Kacharza („Jak wyjść z ramek bocznych") i rysunkiem — schematem ideowym ulepszonego samplera dla Amigi 500/2000. Brakujące materiały zamieszczamy, a Czytelników gorąco przepraszamy.

```
        *= $4000
        sei
        lda #<irq1
        sta $0314
        lda #>irq1
        sta $0315
        lda #$01
        sta $d01a
        lda #$7f
        sta $dc0d
        lda #$31
        sta $d012
        lda #$1b
        sta $d011
        lda #$32
        sta $d00f
        sta $d00d
        sta $d00b
        sta $d009
        lda #$f0
        sta $d015
        lda #$1b
        sta $d011
        lda #<tekst
        ldy #>tekst
        jsr $ab1e
        lda #$80
        sta $d010
        lda #$50
        sta $d00e
        lda #$30
        sta $d00c
        lda #$12
        sta $07fc
        sta $do2c
        cli
nic     jmp nic
;--------------------
irq1    lda #<irq2
        sta $0314
        lda #>irq2
        sta $0315
        cli
petelka nop
        jmp petelka
;--------------------
irq2    ldy #$04
linia1  dey
        bne linia1
        lda #$00
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
;--------------------
        lda $d012
        cmp #$33
        beq aa
aa      ldy #$07
linia2  dey
        bne linia2
        lda #$c0
        sta $d016
        lda #$c8
        sta $d016
;--------------------
        ldy #$07
linia3  dey
        bne linia3
        lda #$00
        nop
        lda #$c0
        sta $d016
        lda #$c8
        sta $d016
;--------------------
        ldy #$07
linia4  dey
        bne linia4
        lda #$00
        nop
        lda #$c0
        sta $d016
        lda #$c8
        sta $d016
;--------------------
        ldy #$07
linia5  dey
        bne linia5
        lda #$00
        nop
        lda #$c0
        sta $d016
        lda #$c8
        sta $d016
;--------------------
        ldy #$07
linia6  dey
        bne linia6
        lda #$00
        nop
        lda #$c0
        sta $d016
        lda #$c8
        sta $d016
;--------------------
        ldy #$07
linia7  dey
        bne linia7
        lda #$00
        nop
        lda #$c0
        sta $d016
        lda #$c8
        sta $d016
;--------------------
        ldy #$07
linia8  dey
        bne linia8
        ldx #$00
        lda #$c8
        ldy #$c0
        sty $d016
        sta $d016,x
;--------------------
linia9  sty $d016
        sta $d016
;--------------------
        ldy #$07
linia10 dey
        bne linia10
        lda #$00
        nop
        nop
        lda #$c0
        sta $d016
        lda #$c9
        sta $d016
;--------------------
        ldy #$07
linia11 dey
        bne linia11
        lda #$00
        nop
        lda #$c0
        sta $d016
        lda #$ca
        sta $d016
;--------------------
        ldy #$07
linia12 dey
        bne linia12
        lda #$00
        nop
        lda #$c0
        sta $d016
        lda #$cb
        sta $d016
;--------------------
        ldy #$07
linia13 dey
        bne linia13
        lda #$00
        nop
        lda #$co
        sta $d016
        lda #$cc
        sta $d016
;--------------------
        ldy #$07
linia14 dey
        bne linia14
        lda #$00
        nop
        lda #$c0
        sta $d016
        lda #$cd
        sta $d016
;--------------------
        ldy #$07
linia15 dey
        bne linia15
        lda #$00
        nop
        lda #$c0
        sta $d016
        lda #$ce
        sta $d016
;--------------------
        ldy #$07
linia16 dey
        bne linia16
        ldx #$00
        lda #$c8
        ldy #$c0
        sty $d016
        sta $d016,x
;--------------------
linia17 sty $d016
        sta $d016
;--------------------
        ldy #$07
linia18 dey
        bne linia18
        lda #$00
        nop
        nop
        lda #$c0
        sta $d016
        lda #$c8
        sta $d016
;--------------------
        ldy #$07
linia19 dey
        bne linia19
        lda #$00
        nop
        lda #$c0
        sta $d016
        lda #$c8
        sta $d016
;--------------------
        ldy #$07
linia20 dey
        bne linia20
        lda #$00
        nop
        lda #$c0
        sta $d016
        lda #$c8
        sta $d016
;--------------------
        ldy #$07
linia21 dey
        bne linia21
        lda #$00
        nop
        lda #$c0
        sta $d016
        lda #$c8
        sta $d016
;--------------------
        nop
        nop
        lda #$06
        sta $d021
;--------------------
        inc $d019
        lda #<irq1
        sta $0314
        lda #>irq1
        sta $0315
        pla
        pla
        pla
        pla
        pla
        pla
        jmp $ea81
;----------------------------
tekst
        .text "0123456789"
        .text "0123456789"
        .text "0123456789"
        .text "0123456789"
        .text ""
        .byte 13,66
        .byte 0
```



SUPERMARKET

Sprzedam Amigę 500 (wersja niemiecka) plus rozszerzenie pamięci do 1 MB wraz z monitorem 1084S na gwarancji plus gry oraz filtr antyradiacyjny na monitor. Cena do uzgodnienia. Sylwester Iwanicki, ul. II Armii W.P 5/6, 68-100 ŻAGAŃ.

Sprzedam C-64 II (gwarancja do czerwca 1993), magnetofon 1530, monitor mono, 3 moduły, joystick, 750 programów, literatura oraz filtr na monitor. Cena kompletu 3.1 mln zł. Krzysztof Starostka, ul. Świerczewskiego 16a, 33-100 TARNÓW, tel. 21-26-17.

Nawiążę kontakty z początkującymi (asembler). Założę grupę (C-64, 1541). Rafał Skórka, Mikołajowice 45, 33-121 BOGUMIŁOWICE, woj. tarnowskie.

Sprzedam oryginalną grę LEANDER firmy PSYGNOSIS dla AMigi. Cena 150000 zł. Patryk Tnópek, Fatimska 10/34, 34-831 KRAKÓW, tel. 43-76-25.

Kuplę opisy do programów: CrossDOS V4.0, SuperCard, Ami-II dla AMigi. Patryk Tnópek, ul. Fatimska 10/34, 31-831 KRAKÓW, tel. 43-76-25.

Sprzedam C-64 II, magnetofon, Black Box 3, 300 gier na kasetach, joystick, literaturę. Cena ok. 2.4 mln. Sebastian Wojcieszek, Zielna 9/100, 26-110 SKARŻYSKO-KAMIENNA, tel. 51-30-96.

Sprzedam za ok. 3.5 mln złotych lub zamienię za dopłatą 1 mln złotych na C-64 komputer AMSTRAD CPC 464 (monitor, klawiaturę z magnetofonem, joystick, 20 kaset, literaturę. Krzysztof Miąsek, Paderewskiego 5/12, 67-400 WSCHOWA, woj. leszczyńskie.

Wymienię lub odkupię programy dla C-64. Poszukuję: Logo Editor (Centauri), F-19 Stealth Fighter, F-15 Strike Eagle, Conflict in Vietnam, Pirates - wszystkie w wersji taśmowej. M. Burkot, ul. Redutowa 2/80, 22-400 ZAMOŚĆ, tel. 75-399.

Zamienię C-64, Black Box 3, joystick, 673 programy i wieżę "Diora" ZGB 110W na gwarancji na Amigę 500 z modulatorem TV. A. Jęch, Oś. Konst. 3-Maja 23/7, 63-200 JAROCIN.

Sprzedam: C-64, magnetofon, moduły, literaturę, oprogramowanie. Robert Strudziński, ul. Królowej Jadwigi 39/53, 33-300 NOWY SĄCZ.

Sprzedam C-64G z magnetofonem, moduł "X", mysz, joystick SV-124 + oprogramowanie (38 kaset z grami i programami użytkowymi), literaturę. Rafał Krawczyk, Lniana 19/30, 91-158 ŁÓDŹ.

Oferuję 700 tys. złotych, Datasette 1530 (gwarancja), 16 kaset (450 programów), walkman M-88 stereo w zamian za używaną stację do C-64. Piotr Cieśla, Czeladzka 39/154, 41-200 SOSNOWIEC.

Posiadam C-64, stację i magnetofon. Wymienię gry i programy użytkowe. Zbigniew Kowalski, Strzelców Byt. 5/5, 45-084 OPOLE, tel 375-77.

Sprzedam stację dysków 5.25" STODOŁA do Amigi oraz 450 nagranych dyskietek. Cena 3.5 mln. Adam Lesień, Hubala 6/28, 94-049 ŁÓDŹ, tel 86-89-64.

Kupię Amigę 500. Piotr Pasternak, Krowoderskich Zuchów 20/1, 31-272 KRAKÓW, tel. 33-82-94.

Sprzedam tanio kasety z grami dla C-64. Sebastian Skowronek, Gałczyńskiego 10/2, 08-300 SOKOŁÓW PODLASKI.

Sprzedam moduł FINAL III do C-64 (250 tys. + porto) opisany w numerze 1 C&A. Rafał Czapiewski, Waryńskiego 4/89, 69-600 CHOJNICE.

Sprzedam C-64 z magnetofonem (gwarancja) + pokrywę na klawiaturę, moduł KERNAL, 15 kaset (w tym 6 z grami), literaturę. Cena 2150000. Jacek Kargól, Skłodowskiej-Curie 38/22, 85-094 BYDGOSZCZ, tel. 41-83-26.

Wymienię, kupię lub sprzedam gry i programy dla C-64. Poszukuję i oferuję opisy. Nośnik taśma lub dyskietka. Krzysztof Skibicki, GDYNIA-KOSAKOWO 30A, 81-198 Gdynia.

Sprzedam książkę "Jak rozbudować interpreter" K. Gajewskiego i B. Radziszewskiego. Krzysztof Apanel, Orla 36/8, 59-300 LUBIN, woj. legnickie.

Wymiana oprogramowania, literatury, doświadczeń - C-64, Star LC-20, C-128. M. Kokoszka, Chopina 7, 57-320 POLANICA.

Kupię magazyn "Amigowiec" nr 4. Sprzedam joystick Quickjoy SV-126 za 100000 zł, książki "Commodore 64" B.Frelka (45-50 tys.) oraz "Poznajemy komputer C-64" (40-45 tys). Mariusz Gaik, 34-615 SŁOMICE 829, woj. Nowy Sącz.

Zamienię C-64C (gwarancja), magnetofon 1530 (gwarancja), Black Box V3, Action Replay V7.3, 2 joysticki, 25 kaset, po- krywę dla C-64, literaturę opraz magnetofon kasetowy MK-250 na C-128D w dobrym stanie (ew. dopłacę). Rajmund Steczeń, ul. Traugutta 25, 87-140 CHEŁMŻA, tel 75-32-84.

Pilnie i tanio sprzedam C-64C + magnetofon + 200 gier + Black Box V4 + 2 joysticki + literatura za około 1900000 zł. Łukasz Zalewski, Meissnera 15, 59-700 BOLESŁAWIEC.

Sprzedam: Bajtki, układ scalony ADC 0809, pomogę w zrobieniu samplera do Amigi lub zrobię na zamówienie. Informacja: koperta + znaczek. Radosław Sapieja, ul. Gliniana 77/1, 50-526 WROCŁAW.

Sprzedam trójnik do dwóch magnetofonów do C-64 za 50000 zł. Info (koperta + znaczek): Roman Dec, Podstawskiego 21, 36-140 SOKOŁÓW MŁP.

Pilnie sprzedam C-128 (stan b.dobry), magnetofon, moduł "X", joystick, pióro świetlne, literaturę oraz oprogramowanie. Cena 2.5 mln. Marek Pacholski, Bitwy nad Bzurą 24/36, 99-100 Łęczyca, tel. 40-59.

Sprzedam C-64, magnetofon, joystick, moduł Black Box, 13 kaset z programami. Cena 2 mln zł. Adam Sydor, Warszawska 32, 62-650 KŁODAWA, woj. konińskie.

Kupię stację 1541-II do C-64. Oferty listowne (opis + cena): Robert Borowski, Orla 42/14, 95-200 PABIANICE.

Wymienię lub odsprzedam programy dla C-64 (dyskietka lub kaseta). Mam około 620 programów i 40 opisów. Krzysztof Szostak. Mrongowiusza 5C, 14-100 OSTRÓDA.

Sprzedam C-64, 1541, 24 kasety, 60 dyskietek z pudełkiem, osłonę na klawiaturę, 3 moduły (TURBO, 3-SUPER GAME, ACTION REPLAY V7), 2 magnetofony, 2 joysticki, literaturę i schematy za 4 mln zł. Dawid Góral, Oś.Zielone 16/20, 33-100 TARNÓW, tel 24-01-42.

Sprzedam C-64-II, magnetofon, 2 joysticki (Commander) (na gwarancji), moduł Turbo Tape Master, 450 programów, literaturę za 1.9 mln. Rafał Muzyk, Gdańska 6b/10, 84-300 LĘBORK.

Kupię stację 1541-II do 2 mln złotych oraz monitor 1082D lub 1084S. Maciej Syguła, Mielczarskiego 10/1, 95 OZORKÓW, tel. 18-20-07.

Sprzedam C-64, magnetofon, joysticki, moduł PLUS, około 1500 programów, całość ma 3 miesiące. Cena 2 mln złotych. Kupię Amigę 500 lub Atari ST. Mariusz Rus, Krótka 2, 32-350 PILICA.

Wymienię gry i programy dla C-64 na dyskietkach. Poszukuję F-19 Stealth Fighter. Łukasz Nosal. Złota 11a/8, 78-100 KOŁOBRZEG, tel 216-28.

Sprzedam C-64II, stację 1541, programy (GEOS V1.3 z instrukcją po polsku, GIGA CAD V1.0, Fast Hack'em V4.0, Voicetracker, Future Composer, 15 gier, joystick Supercharger, literaturę dla C-64 i 1541. Gwarancja do 15-01-1993. Cena kompletu: 3.8 mln. Mariusz Koczupinda, Polna 6, 48-120 BABORÓW, woj. opolskie.

Odkupię lub pożyczę mapę pamięci do Commodore 64. Artur Czekuliński, Fornalskiej 20/5, 55-231 JELCZ - LASKOWICE, woj. wrocławskie.

Sprzedam C-64II, stację 1541-II, magnetofon, pokrywę, moduły Final III i Black Box, dyskietki za 3.7 mln. Wojciech Ogrodniczak, KRN 5/34, 22-100 CHEŁM, tel 576-36 po 1500.

Sprzedam gry dla C-64. Wykonam zasilacze bardziej wytrzymałe i unowocześnione z sygnalizacją działania (diody) i przeciążenia). Poszukuję płyty C-64 i magnetofonu. Jacek Gałkowski, Ks. Krausego 17/21, 86-105 ŚWIECIE.

Zamienię C-64, stację 1541-II, magnetofon, Final II, Ex-plus, Joy-Turbo II, 25 dyskietek, 12 kaset 6 książek, aparat fot. "Wilia" na używaną Amigę 500. M.Kajstura, Jaśminowa 17, 43-410 KOŃCZYCE M.

Sprzedam C-128 (stan idealny), stację dysków 9900 (na gwarancji), magnetofont, joysticki, moduły Final III i SUPERGAME, 13 kaset, 20 dyskietek oraz literaturę. Cena do ustalenia (ok. 5 mln). Radosław Boroński, Karłowicza 3/27, 62-510 Konin Zatorze.

COMMODORE

— krótka historia sukcesu

Każdy, kto ma do czynienia z komputerami, z pewnością zna te nazwy: Amiga 500, Amiga 3000, C-64 czy CDTV (no, może tę ostatnią jeszcze nie wszyscy). Natomiast nazwy takie jak PET 2001 lub VC 20 przechodzą powoli w zapomnienie, mimo że w latach siedemdziesiątych stanowiły synonim komputerów dobrej jakości. To właśnie dzięki tym urządzeniom firma Commodore, założona w roku 1958 w Toronto jako „Commodore Portable Typewriter Company", weszła do grona światowych potentatów w dziedzinie elektroniki.

Właściwa historia firmy zaczyna się jednak na dobre w roku 1960, kiedy to w Westchester w Pensylwanii powstała placówka określająca się mianem „Commodore Business Machines Inc." Wówczas, najpierw na rynku amerykańskim, stworzyła się luka: coraz więcej biur i przedsiębiorstw potrzebowało na gwałt czegoś, co zastąpiłoby dotychczasowe przechowywanie informacji na papierze. Banki, giełdy, różnorakie jednostki administracyjne gubiły się w ciągle rosnącym potoku nowych danych, nie nadążały z ich usystematyzowaniem i przetwarzaniem. Dopiero dzięki tej hossie na wszelkie maszyny ułatwiające liczenie, archiwizację i inne tego rodzaju operacje konieczne do prowadzenia działalności gospodarczej, firmy takie jak IBM czy Commodore mogły zaistnieć na rynku i przestać obawiać się o swoją egzystencję. Problemy ze znalezieniem klienta skończyły się raz na zawsze (a przynajmniej nie było ich do dzisiaj).

W roku 1967 Commodore wprowadziła na rynek pierwszy ręczny, specjalizowany kalkulator umożliwiający stałą aktualizację danych (np. stan konta w banku). W celu dalszego rozwoju tej rewolucyjnej, jak na owe czasy technologii, kierownictwo firmy stworzyło w legendarnej kalifornijskiej Silicon Valley własne laboratorium (1968).

Stało się ono kolebką komputerów osobistych Commodore, zaprezentowanych po raz pierwszy w 1977 na National Computer Conference w Dallas. PET 2001 — praszczur dzisiejszych pecetów — ze swoją czterokilobajtową pamięcią okrzyknięty został wtedy prawdziwą rewelacją. Prócz „ogromnej" pamięci i sporej szybkości obliczeń cechowało go jeszcze jedno: był bardzo tani. Rzeczywiście, jak na ówczesne warunki, 595 dolarów za takie cudeńko stanowiło wyjątkowo korzystną cenę. Zresztą producent do dzisiaj jest wierny zasadzie: „Lepsza technika, niska cena".

W latach następnych rozwój informatyki przybrał szaleńcze tempo. Firma Commodore dostrzegła w porę, iż komputer nie musi służyć wyłącznie celom profesjonalnym. Wprowadziła zatem na rynek VC 20, który wręcz podbił rynek światowy zwłaszcza dzięki swym możliwościom graficznym. Do roku 1982 sprzedano na całym świecie ponad milion komputerów tego typu.

Jednak prawdziwym przebojem, który pobił wszelkie rekordy, stał się dobrze wszystkim znany Commodore C-64. Wyprodukowany w 1982 roku, z założenia miał być komputerem dla każdego i to przesądziło o jego popularności. Służył równie dobrze dzieciom do zabawy (jak do tej pory rekordzista w oprogramowaniu jeśli chodzi o gry) jak i — w swoim czasie — przedsiębiorstwom do rachunkowości czy uniwersytetom do nauczania. Zresztą do dzisiaj, z powodu swych znakomitych możliwości graficznych, dźwiękowych i łatwości obsługi C-64 cieszy się zrozumiałym uznaniem i jest niezastąpionym produktem dla początkujących entuzjastów informatyki. Tylko w latach 1990/91 sprzedano więcej niż 400 000 tych komputerów, zaś ogółem ponad 10 milionów (!). Co by nie mówić, to naprawdę imponujący sukces!



„Staruszek" C-64 ...



... i AMIGA 3000T — olbrzymi postęp w krótkim czasie.

Po roku 1985 firma Commodore zajęła się konstruowaniem własnych klonów komputerów PC. Obecnie na europejskim rynku pecetów, według angielskiego instytutu badań rynkowych Dataquest, Commodore zajmuje drugie miejsce (jeśli chodzi o ilość sprzedanych maszyn), natomiast w USA (według rankingów amerykańskich) — piąte.

Kolejnym produktem była Amiga. Ten komputer ma ciągle szanse stać się jeszcze większym szlagierem rynkowym niż „staruszek C-64". Przyjęta za czasów PET-a 2001 maksyma — „komputer to niekoniecznie urządzenie dla celów profesjonalnych" — sprawdza się do dziś i przysparza producentowi Amigi stale wielu nowych zwolenników. Przypuszcza się, że gdzieś pod koniec 1992 roku zostanie sprzedana 3,5-milionowa Amiga (w samych tylko Niemczech sprzedano już ponad milion tych komputerów). Powodem tak wielkiej popularności tego cacka są oczywiście wspaniałe możliwości graficzne i dźwiękowe pozwalające na wykorzystanie go równie dobrze do rozrywki jak i do pracy. Magazyn „Chip" i niemieccy specjaliści wybrali Amigę 500 na „komputer roku" (1991), a międzynarodowe jury już po raz drugi na „europejski komputer roku" (tytuł ten to swego rodzaju „komputerowy Oscar").

Do chwili obecnej firma Commodore rozrosła się do ogromnych rozmiarów. Posiada szereg filii zagranicznych, jej wyroby znane są na całym świecie. W roku 1990 zreorganizowała się i dzieli się na cztery zasadnicze jednostki: dział pecetów, dział sieci komputerowych, dział Amigi oraz dział do spraw użytkowników (chodzi tu głównie o domowe zastosowania komputerów).

Uruchomiona w 1991 nowa linia produkcyjna „Profi-Line", co można przełożyć na polski jako „seria profesjonalna", wytwarza najwyższej jakości komputery klasy PC, począwszy od SL 286 a skończywszy na superszybkiej, niezwykle wydajnej wieży T486-25C. W ofercie nie brak też *laptopów* — Notebook C 386 SX-LT jako pierwszy przenośny pecet ma wbudowany modem.

Innym przykładem potwierdzającym dobrą renomę firmy Commodore może być CDTV (Commodore Dynamic Total Vision). Ten multimedialny system wykorzystuje płyty kompaktowe jako nośnik pamięci (na jednej płycie mieści się 550 MB!), a jego sercem jest sprawdzona Amiga 500. CDTV niewątpliwie otworzy nową epokę rozwoju techniki komputerowej a przynajmniej skieruje ją na nowe tory — już od dłuższego czasu zachodnie pisma specjalistyczne nie mogą wyjść z podziwu i określają tę maszynkę jako absolutnie rewelacyjną, genialne wręcz masmedium.

Wydaje się, że firma Commodore trafnie obrała strategię produkcyjną i pewnie stoi na rynkach światowych. Miejmy więc nadzieję, że taki stan rzeczy utrzyma się jeszcze długo — wówczas będziemy mogli do woli cieszyć się i podziwiać wspaniałe możliwości komputerów ze znaczkiem C=, bez których świat byłby z pewnością uboższy.

na podst. „COMM" opracował

ARNOLD